PHẦN VĂN BẢN QUY PHẠM PHÁP LUẬT

BỘ QUỐC PHÒNG

Thông tư số 161/2016/TT-BQP ngày 21 tháng 10 năm 2016 ban hành Quy chuẩn kỹ thuật quốc gia về mật mã dân sự sử dụng trong lĩnh vực ngân hàng

(Tiếp theo Công báo số 1177 + 1178)

QCVN 4:2016/BQP

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ MÃ HÓA DỮ LIỆU SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on data encryption used in banking

2.3.2. Chế độ phản hồi mã CFB (Cipher FeedBack)

Chế độ CFB được xác định bởi 3 tham số:

- Kích thước của bộ đệm phản hồi, r, trong đó $n \leq r \leq 1024n$ và $r < qn$
- Kích thước của biến phản hồi, k, trong đó $1 \leq k \leq n$
- Kích thước của biến bản rõ, j, trong đó $1 \leq j \leq k$

Trong Quy chuẩn này thì giá trị của j và k bằng nhau.

Các biến được sử dụng trong chế độ CFB:

a) Các biến đầu vào

1) Dãy q biến bản rõ độ dài j bit $P_1, P_2, \ldots, P_q$.

2) Khóa bí mật K.

3) Biến khởi đầu độ dài r bit.

b) Các kết quả trung gian:

1) Dãy q khối đầu vào của mã khối độ dài n bit $X_1, X_2, \dots, X_q$.

2) Dãy q khối đầu ra của mã khối độ dài n bit $Y_1, Y_2, \dots, Y_q$.

3) Dãy q biến độ dài j bit $E_1, E_2, \dots, E_q$.

4) Dãy $q - 1$ biến phản hồi độ dài k bit $F_1, F_2, \dots, F_{q-1}$.

5) Dãy q nội dung bộ đệm phản hồi độ dài r bit $FB_1, FB_2, \dots, FB_q$.

c) Các biến đầu ra là dãy q biến bản mã độ dài j bit $C_1, C_2, \dots, C_q$.

2.3.2.1. Phép mã hóa

Bộ đệm phản hồi FB được gán giá trị khởi đầu.

$$FB_1 = SV$$

Phép toán mã hóa mỗi biến bản rõ thực hiện trong 6 bước sau:

a) $X_i = n{\sim}FB_i$ (Lựa chọn n bit tận cùng bên trái của FB).

b) $Y_i = e_K(X_i)$ (Sử dụng mã khối).

c) $E_i = j{\sim}Y_i$ (Lựa chọn j bit tận cùng bên trái của Y_i).

d) $C_i = P_i \oplus E_i$ (Tạo biến bản mã).

e) $F_i = I(k - j)|C_i$ (Tạo biến phản hồi).

f) $FB_{i+1} = S_k(FB_i|F_i)$ (Hàm dịch chuyển trên FB).

Các bước trên lặp với biến đếm $i = 1, 2, \dots, q$, trong vòng lặp cuối dừng lại tại bước (d). j bit tận cùng bên trái của khối đầu ra Y của mã khối được sử dụng để mã hóa j-bit biến bản rõ theo phép cộng modulo 2. Các bit còn lại của Y được bỏ đi. Các bit biến bản rõ/bản mã có chỉ số từ 1 đến j.

Biến bản mã được tăng thêm bằng cách đặt $k - j$ bit trong vị trí bit tận cùng bên trái của mình để tạo biến phản hồi F độ dài k-bit. Sau đó các bit của bộ đệm phản hồi FB được dịch trái đi k vị trí và F được thêm vào vị trí k tận cùng bên phải để tạo ra giá trị mới của bộ đệm phản hồi FB. Trong phép toán dịch, k bit tận cùng bên trái của FB được bỏ đi. n bit tận cùng bên trái mới của FB được sử dụng như là đầu vào tiếp theo của X trong quá trình mã hóa.

2.3.2.2. Phép giải mã

Bộ đệm phản hồi FB được gán giá trị khởi đầu.

$$FB_1 = SV$$

Phép toán giải mã mỗi biến bản mã thực hiện trong 6 bước sau:

a) $X_i = n{\sim}FB_i$ (Lựa chọn n bit tận cùng bên trái của FB).

b) $Y_i = e_K(X_i)$ (Sử dụng mã khối).

c) $E_i = j{\sim}Y_i$ (Lựa chọn j bit tận cùng bên trái của Y_i).

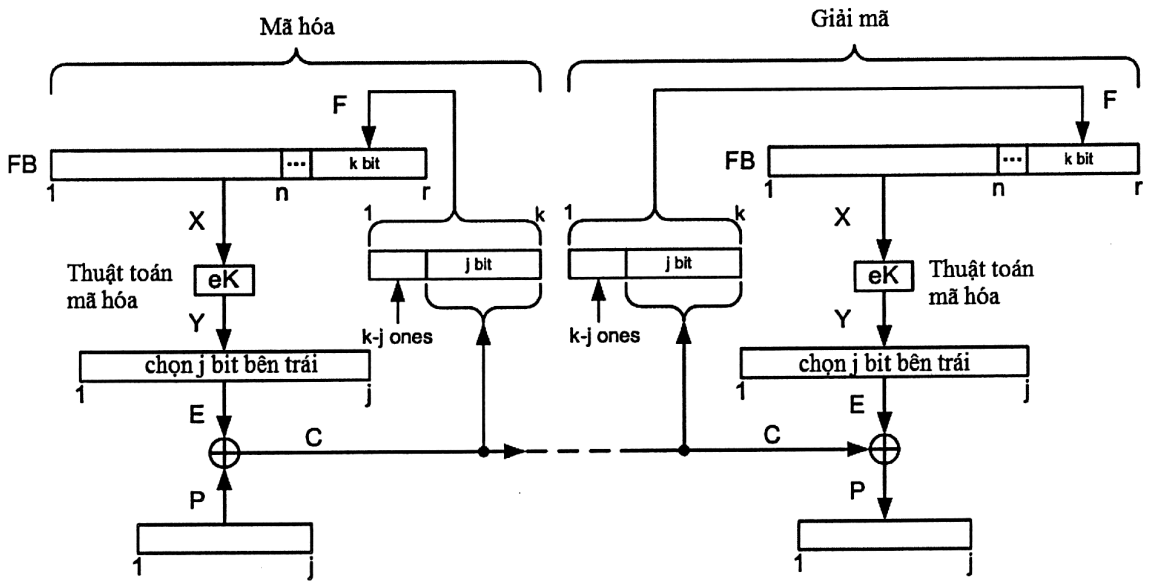
d) $P_i = C_i \oplus E_i$ (Tạo biến bản rõ).

e) $F_i = I(k - j)|C_i$ (Tạo biến phản hồi).

f) $FB_{i+1} = S_k(FB_i|F_i)$ (Hàm dịch chuyển trên FB).

Các bước trên lặp với biến đếm $i = 1, 2, \dots, q$, trong vòng lặp cuối dừng lại tại bước (d). j bit tận cùng bên trái của khối đầu ra Y của mã khối được sử dụng để giải mã j-bit biến bản mã theo phép cộng modulo 2. Các bit còn lại của Y được bỏ đi. Các bit biến bản rõ/bản mã có chỉ số từ *1* đến j.

Biến bản mã được tăng thêm bằng cách đặt $k - j$ bit trong vị trí bit tận cùng bên trái của mình để tạo biến phản hồi F độ dài k-bit. Sau đó các bit của bộ đệm phản hồi FB được dịch trái đi k vị trí và F được thêm vào vị trí k tận cùng bên phải để tạo ra giá trị mới của bộ đệm phản hồi FB. Trong phép toán dịch, k bit tận cùng bên trái của FB được bỏ đi. n bit tận cùng bên trái mới của FB được sử dụng như là đầu vào tiếp theo của X trong quá trình giải mã.



Hình 10: Chế độ phản hồi mã CFB

2.3.3. Chế độ phản hồi đầu ra OFB (Output Feedback):

Chế độ OFB được xác định bằng một tham số j là kích thước biến bản rõ với $1 \le j \le n$.

Các biến được sử dụng trong chế độ OFB là:

a) Các biến đầu vào:

1) Dãy q biến bản rõ độ dài j bit $P_1, P_2, \dots, P_q$.

2) Khóa bí mật K.

3) Biến khởi đầu độ dài n bit.

b) Các kết quả trung gian:

1) Dãy q khối đầu vào của mã khối độ dài n bit $X_1, X_2, \dots, X_q$.

2) Dãy q khối đầu ra của mã khối độ dài n bit $Y_1, Y_2, \dots, Y_q$.

3) Dãy q biến độ dài j bit $E_1, E_2, \dots, E_q$.

c) Các biến đầu ra, nghĩa là dãy q biến bản mã độ dài j bit $C_1, C_2, \dots, C_q$.

2.3.3.1. Phép mã hóa

Khối đầu vào X được gán giá trị khởi đầu

$$X_1 = SV$$

Phép toán mã hóa mỗi biến bản rõ thực hiện trong 4 bước sau:

a) $Y_i = e_k(X_i)$ (Sử dụng mã khối).

b) $E_i = j{\sim}Y_i$ (Chọn j bit tận cùng bên trái).

c) $C_i = P_i \oplus E_i$ (Tạo biến bản mã).

d) $X_{i+1} = Y_i$ (Phép toán phản hồi).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit với chỉ số từ *1* đến j.

Kết quả của mỗi lần sử dụng mã khối là Y_i và được đưa trở lại thành giá trị tiếp theo của X, đặt là X_{i+1}. j bit tận cùng bên trái của Y_i được sử dụng để mã hóa biến đầu vào.

2.3.3.2. Phép giải mã

Khối đầu vào *X* được gán giá trị khởi đầu

$$X_1 = SV$$

Phép toán giải mã mỗi biến bản mã thực hiện trong 4 bước sau:

a) $Y_i = e_k(X_i)$ (Sử dụng mã khối).

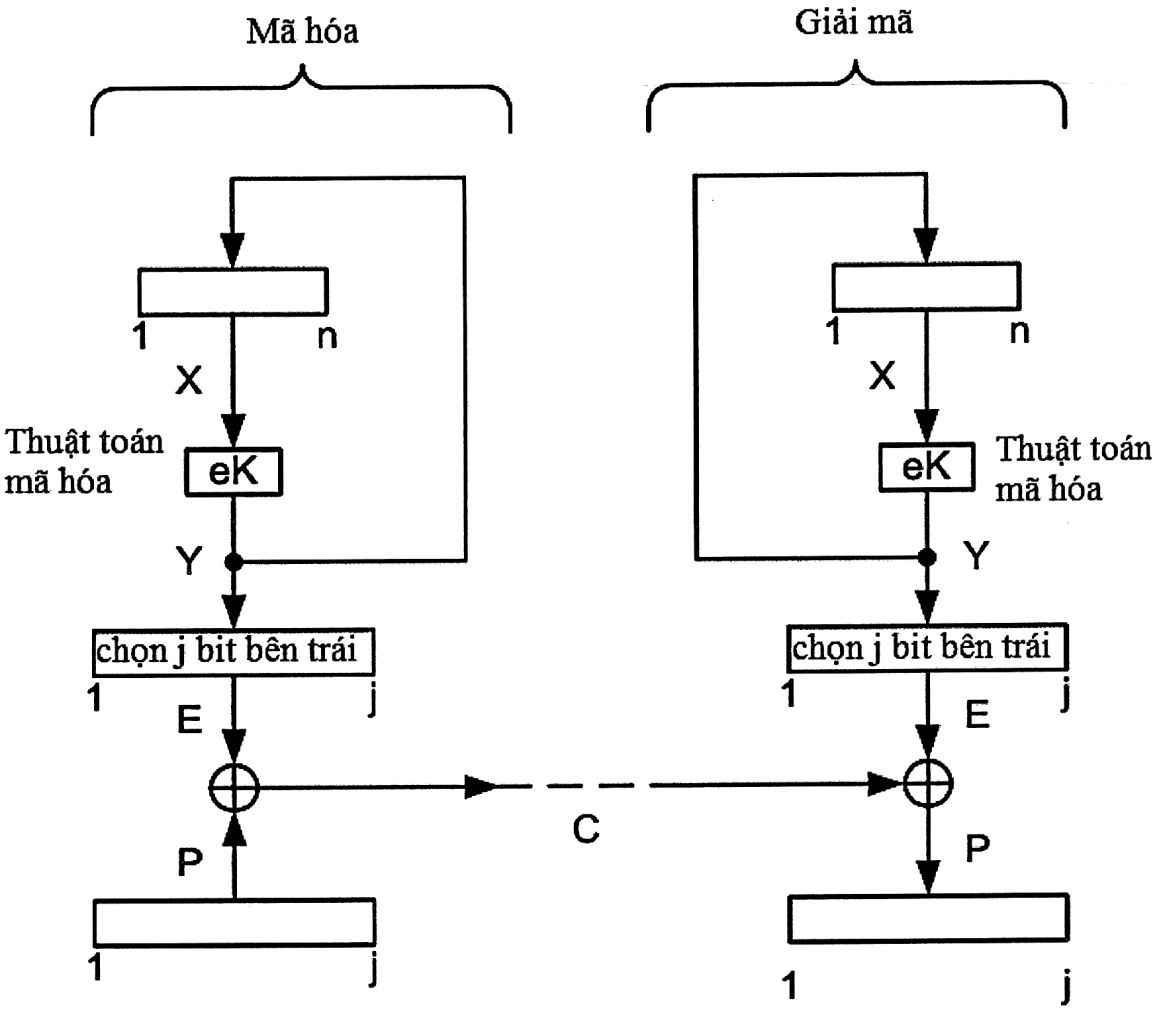
b) $E_i = j{\sim}Y_i$ (Chọn j bit tận cùng bên trái).

c) $P_i = C_i \oplus E_i$ (Tạo biến bản mã).

d) $X_{i+1} = Y_i$ (Phép toán phản hồi).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit với chỉ số từ *1* đến j.

Kết quả của mỗi lần sử dụng mã khối là Y_i và được đưa trở lại thành giá trị tiếp theo của *X*, đặt là X_{i+1}. j bit tận cùng bên trái của Y_i được sử dụng để giải mã biến đầu vào.



Hình 11: Chế độ phản hồi đầu ra OFB

2.3.4. Chế độ đếm CTR (Counter)

Chế độ CTR được xác định bằng một tham số j trong đó $1 \leq j \leq n$.

Các biến được sử dụng trong chế độ *CTR* là:

a) Các biến đầu vào:

1) Dãy q biến bản rõ $P_1, P_2, \dots, P_q$, mỗi biến có độ dài j bit.

2) Khóa bí mật K.

3) Biến khởi đầu SV có độ dài n bit.

b) Các kết quả trung gian:

1) Dãy q khối đầu vào của mã khối có độ dài n bit $CTR_1, CTR_2, \dots, CTR_q$.

2) Dãy q khối đầu ra của mã khối có độ dài n bit $Y_1, Y_2, \dots, Y_q$.

3) Dãy q biến có độ dài j bit $E_1, E_2, \dots, E_q$.

c) Các biến đầu ra, nghĩa là dãy q biến bản mã độ dài j bit $C_1, C_2, \dots, C_q$.

2.3.4.1. Phép mã hóa

CTR được gán giá trị khởi đầu

$$CTR_1 = SV$$

Các phép toán để mã hóa mỗi biến bản rõ theo 4 bước sau:

a) $Y_i = e_K(CTR_i)$ (Sử dụng mã khối).

b) $E_i = j{\sim}Y_i$ (Chọn j bit tận cùng bên trái của Y_i).

c) $C_i = P_i \oplus E_i$ (Tạo ra biến bản mã).

d) $CTR_{i+1} = (CTR_i + 1) mod\ 2^n$ (Tạo ra giá trị đếm mới CTR).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit có chỉ số từ *1* đến j.

Giá trị đếm được mã hóa để đưa ra khối đầu ra Y_i và j bit tận cùng bên trái của khối đầu ra Y_i được sử dụng để mã hóa giá trị đầu vào. CTR sau đó được tăng 1 ($modulo\ 2^n$) để tạo ra giá trị đếm mới.

2.3.4.2. Phép giải mã

CTR được gán giá trị khởi đầu

$$CTR_1 = SV$$

Các phép toán để giải mã mỗi biến bản rõ theo 4 bước sau:

a) $Y_i = e_K(CTR_i)$ (Sử dụng mã khối).

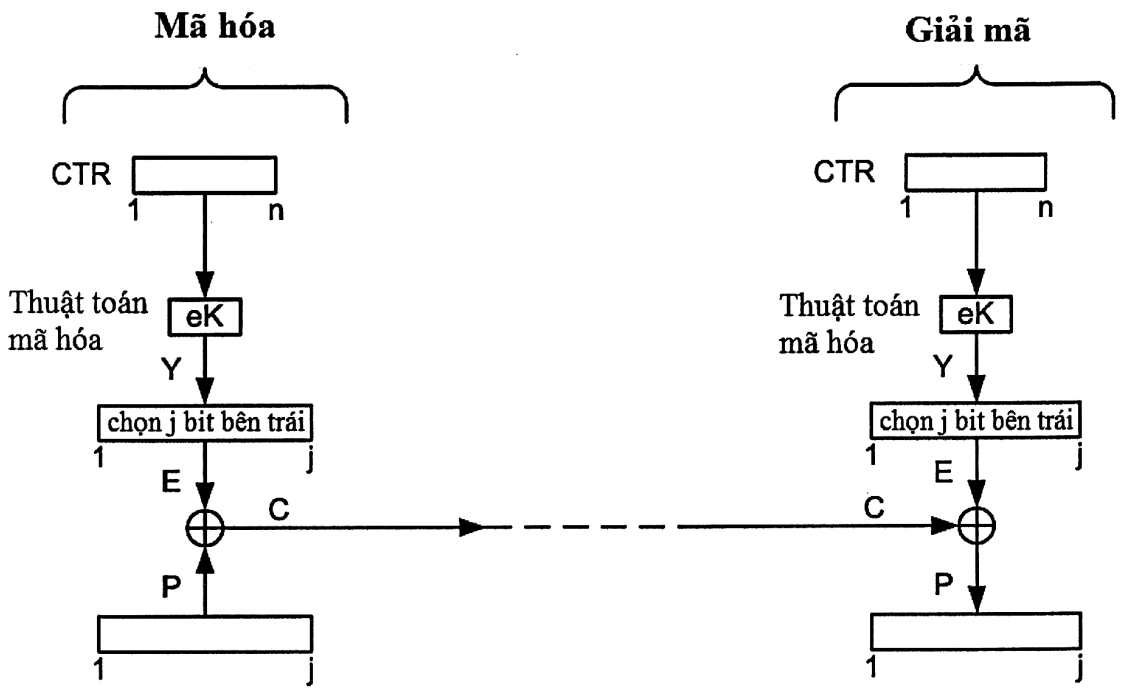
b) $E_i = j{\sim}Y_i$ (Chọn j bit tận cùng bên trái của Y_i).

c) $P_i = C_i \oplus E_i$ (Tạo ra biến bản rõ).

d) $CTR_{i+1} = (CTR_i + 1)\ mod\ 2^n$ (Tạo ra giá trị đếm mới CTR).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit có chỉ số từ *1* đến j.

Giá trị đếm được mã hóa để đưa ra khối đầu ra Y_i và j bit tận cùng bên trái của khối đầu ra Y_i được sử dụng để mã hóa giá trị đầu vào. CTR sau đó được tăng 1 (modulo 2^n) để tạo ra giá trị đếm mới.



Hình 12: Chế độ đếm CTR

2.4. Mã dòng

Mã dòng là mã thực hiện biến đổi từng bit dữ liệu của bản rõ sang bản mã sử dụng khóa độ dài k bit. Quy chuẩn này quy định sử dụng các mã khối được quy định tại Mục 2.2 sử dụng chế độ CFB, OFB hoặc CTR để thực hiện chức năng mã dòng.

3. QUY ĐỊNH VỀ QUẢN LÝ

3.1. Các mức giới hạn của đặc tính kỹ thuật mật mã và yêu cầu quản lý của các thuật toán mật mã để mã hóa dữ liệu nêu tại Quy chuẩn này là các chỉ tiêu chất lượng phục vụ được quản lý theo quy định về quản lý chất lượng sản phẩm, dịch vụ mật mã dân sự được quy định tại Luật An toàn thông tin mạng ngày 19 tháng 11 năm 2015.

3.2. Hoạt động kiểm tra chất lượng sản phẩm, dịch vụ mật mã được cơ quan quản lý nhà nước có thẩm quyền tiến hành định kỳ hàng năm hoặc đột xuất.

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

4.1. Các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự có trách nhiệm đảm bảo tuân thủ Quy chuẩn này và chịu sự kiểm tra của cơ quan quản lý nhà nước theo quy định.

4.2. Doanh nghiệp cung cấp sản phẩm, dịch vụ mật mã dân sự cho các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) có trách nhiệm thực hiện công bố hợp quy sản phẩm, dịch vụ mật mã dân sự phù hợp với Quy chuẩn này. Việc công bố hợp quy thực hiện theo Thông tư số 28/2012/TT-BKHCN ngày 12 tháng 12 năm 2012 của Bộ Khoa học và Công nghệ.

4.3. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ có trách nhiệm tiếp nhận đăng ký công bố hợp quy, thực hiện quản lý, hướng dẫn và kiểm tra việc công bố hợp quy.

5. TỔ CHỨC THỰC HIỆN

5.1. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ có trách nhiệm hướng dẫn, tổ chức triển khai quản lý kỹ thuật mật mã của thuật toán mã hóa dữ liệu theo Quy chuẩn này.

5.2. Trong trường hợp các quy định nêu tại Quy chuẩn kỹ thuật quốc gia này có sự thay đổi, bổ sung hoặc được thay thế thì thực hiện theo quy định tại văn bản mới./.

Phụ lục A

(Quy định)

Mô tả DES

A.1. Mở đầu

Thuật toán DES là mã khối đối xứng có thể xử lí các khối dữ liệu 64 bit, sử dụng khóa bí mật độ dài 64 bit. Mỗi bit thứ tám của khóa mật mã thường được sử dụng để kiểm tra tính chẵn lẻ và được bỏ qua.

A.2. Phép mã hóa DES

Phép mã hóa được chỉ ra trên Hình A.1

Bản rõ 64-bit trước hết được biến đổi qua hoán vị ban đầu ***IP***. Sau đó khối được chia thành hai nửa $\boldsymbol{L_0}$ và $\boldsymbol{R_0}$, mỗi nửa gồm 32 bit. Tiếp đó thực hiện 16 vòng biến đổi giống nhau được gọi là hàm f, trong đó dữ liệu được kết hợp với khóa. Trong mỗi vòng, nửa phải là đầu vào của hàm f được khóa hóa, hàm này nhận đầu vào 32 bit và khóa con 48 bit $\boldsymbol{K_i}$ và cho đầu ra 32-bit. Đầu ra này tiếp đó được XOR với nửa trái để tạo ra nửa trái mới đã được biến đổi. Tại phần cuối của mỗi vòng, trừ vòng cuối cùng, hai nửa trái và phải đổi chỗ cho nhau để tạo ra $\boldsymbol{L_i}$ và $\boldsymbol{R_i}$ tương ứng. Sau khi thực hiện vòng cuối cùng, hai nửa trái và phải được ghép lại với nhau và khối 64-bit nhận được lại được biến đổi qua phép hoán vị cuối $\boldsymbol{IP^{-1}}$ là hoán vị nghịch đảo của hoán vị ban đầu ***IP***. Đầu ra là bản mã 64-bit.

Phép mã hóa được xác định như sau (P và C là dữ liệu, K_i là khóa).

(1) $IP(P) = L_0 \parallel R_0$

(2) Với $i = 1, 2, \ldots, 16$:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

(3) $C = IP^{-1}(R_{16} \parallel L_{16})$

A.3. Phép giải mã DES

Phép giải mã cũng giống như phép mã hóa. Sự khác nhau chỉ ở chỗ, các khóa con K_i được sử dụng theo thứ tự ngược lại.

A.4. Các hàm DES

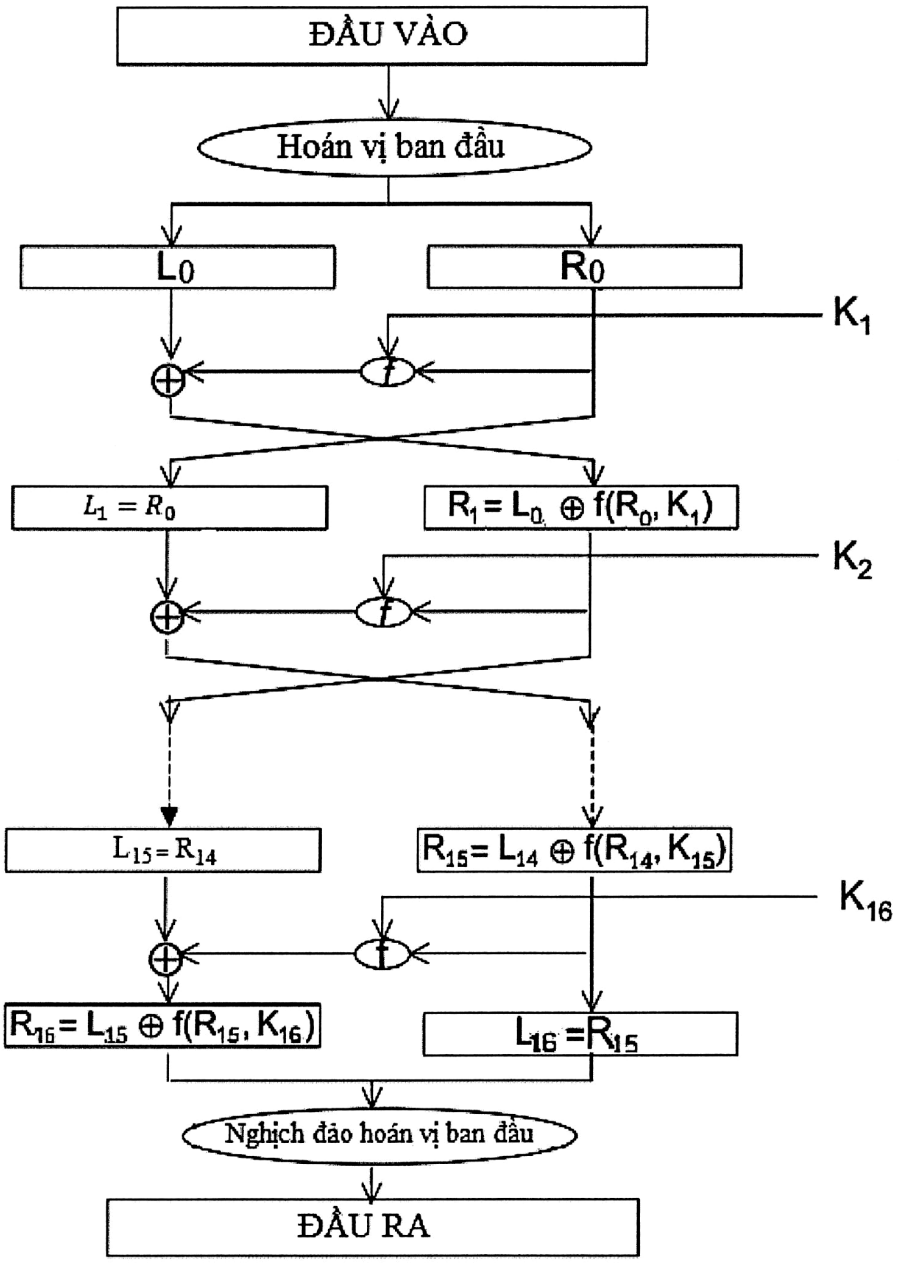
A.4.1. Phép hoán vị ban đầu IP

Phép hóa vị ban đầu $\boldsymbol{IP}$ được chỉ ra tại Bảng A.1. Hoán vị này nhận đầu vào 64-bit và cho đầu ra 64-bit, theo đó bit thứ nhất được hoán vị thành bit thứ 58, bit thứ hai thành bit thứ 50,v.v, bit cuối cùng thành bit thứ 7.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3

61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Bảng A.1 - Hoán vị khởi tạo



Hình A.1 - Thủ tục mã hóa

A.4.2. Hoán vị nghịch đảo IP^{-1}

Hoán vị nghịch đảo IP^{-1} được chỉ ra trên Bảng A.2, nhận đầu vào 64-bit và cho đầu ra 64-bit. Đầu ra của Thuật toán nhận bit thứ 40 của khối đầu ra trước đó làm bit thứ nhất, bit thứ 8 làm bit thứ hai,v.v và bit thứ 25 của khối đầu ra trước đó làm bit cuối cùng.

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Bảng A.2 - Hoán vị nghịch đảo IP^{-1}

A.4.3. Hàm *f*

Hàm ***f*** được chỉ ra trên Hình A.2

Hàm ***f*** nhận đầu vào 32 bit R và mở rộng thành R' có độ dài 48-bit bằng cách sử dụng phép hoán vị mở rộng $\boldsymbol{E}$. Sau đó 48-bit R' được XOR với khóa con 48-bit K, thu được dữ liệu 48-bit, dữ liệu này được viết thành 8 khối, mỗi khối 6-bit, bằng cách chọn các bit đầu vào của nó theo một thứ tự được quy định trên bảng. Các hàm được chọn duy nhất đó được gọi là các $\mathrm{S-box}$, $\mathbf{S_1, S_2, \ldots, S_8}$, nhận đầu vào là các khối 6-bit r_i và cho đầu ra là các khối 4-bit $S_i(r_i)$. Hàm hoán vị $\boldsymbol{P}$ cho đầu ra 32-bit R'''' từ đầu vào 32-bit R''' bằng cách hoán vị các bit của khối đầu vào. R'''' là đầu ra của hàm ***f***.

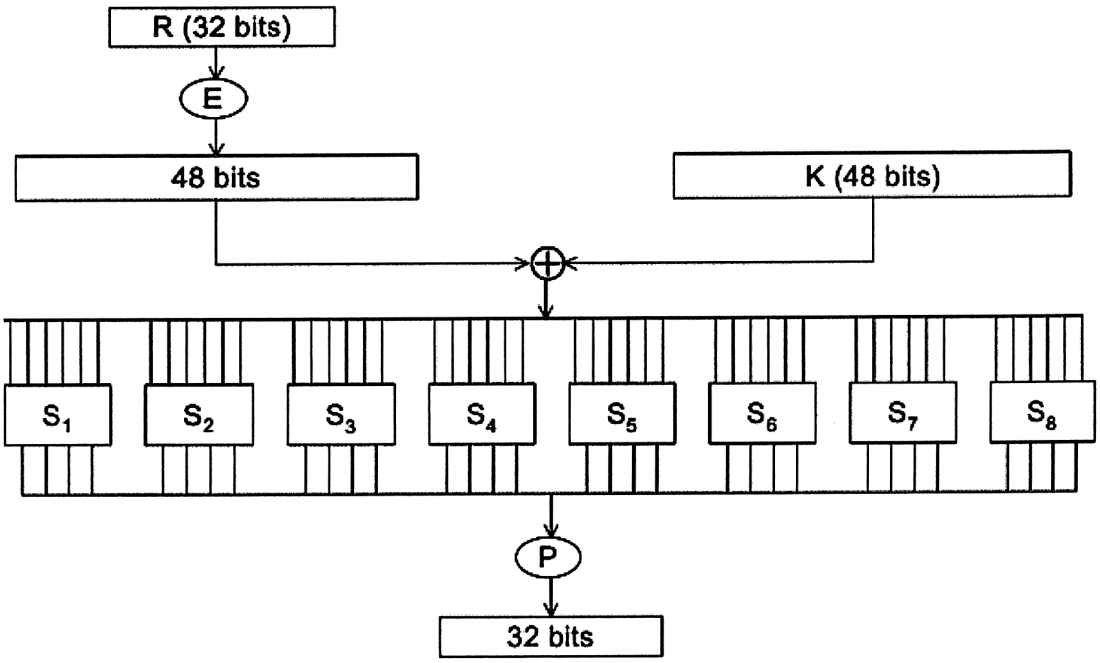
Hàm ***f*** do đó được xác định như sau (P và C là dữ liệu, K_i là khóa)

(1) $R' = E(R)$

(2) $R'' = R' \oplus K$

(3) $R'' = r1 \parallel r2 \parallel r3 \parallel r4 \parallel r5 \parallel r6 \parallel r7 \parallel r8$

(4) $R'''' = P(R''')$



Hình A.2 - Tính $f(R, K)$

A.4.4. Hoán vị mở rộng E

Hoán vị mở rộng E được chỉ ra trên Bảng A.3. Hoán vị E nhận đầu vào 32- bit và cho đầu ra 48-bit. Ba bit đầu tiên của E là các bit ở các vị trí 32, 1 và 2, hai bit cuối ở các vị trí 32 và 1.

32	1	2	3	4	5	64	32
4	5	6	7	8	9	63	31
8	9	10	11	12	13	62	30
12	13	14	15	16	17	61	29
16	17	18	19	20	21	60	28
20	21	22	23	24	25	59	27
24	25	26	27	28	29	58	26
28	29	30	31	32	1	57	25

Bảng A.3 – Hoán vị mở rộng E

A.4.5. Hoán vị P

Hoán vị P được chỉ ra trên Bảng A.4. Hoán vị P nhận đầu vào 32-bit và cho đầu ra 32-bit. Đầu ra $P(L)$ của hàm P được xác định bởi Bảng A.4, thu được từ đầu vào L bằng cách lấy bit thứ 16 của L làm bit thứ nhất của $P(L)$, bit thứ bảy làm bit thứ hai của $P(L)$,v.v và bit thứ 25 của L làm bit thứ 32 của $P(L)$.

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Bảng A.4 – Hoán vị P

A.4.6. S-box

S-box được chỉ ra trên Bảng A.5. Mỗi S-box có 6-bit đầu vào và 4-bit đầu ra.

Nếu S_1 là hàm được xác định trên bảng và ***B*** là khối 6-bit, thì $S_1(B)$ được xác định như sau: các bit đầu tiên và cuối cùng của ***B*** được biểu diễn theo cơ số 2 là một số nằm trong khoảng từ 0 đến 3. Giả sử số đó là i. Bốn bit giữa của ***B*** được biểu diễn theo cơ số 2 là số nằm trong khoảng từ 0 đến 15. Giả sử số đó là j. Trên bảng đó là số nằm trên hàng thứ *i* và cột thứ *j*. Đó là số nằm trong khoảng từ 0 đến 15, và được biểu diễn duy nhất bởi khối 4-bit. Khối

này là đầu ra $S_1(B)$ của S_1 ứng với đầu vào ***B***. Ví dụ, với đầu vào 011011, hàng được biểu diễn bởi 01 là hàng 1, cột được xác định bởi 1101 là cột 13. Nằm trên giao của hàng 1 và cột 13 là số 5, do đó đầu ra là 0101.

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6

4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Bảng A.5 - Các S-box

A.5. Lược đồ tạo khóa DES

Phần lược đồ khóa DES được chỉ ra trên Hình A.3. Lược đồ này nhận khóa 64-bit khóa **KEY** và cho ra 16 khóa con 48-bit $K_1, K_2, \ldots\ldots, K_{16}$.

Trong đó $\boldsymbol{K_n}$, với $1 \leq n \leq 16$ là khối 48-bit trong bước (2) của thuật toán. Bởi vậy để mô tả ***KS*** chỉ cần mô tả việc tính toán của $\boldsymbol{K_n}$ từ ***KEY*** với n = 1, 2..., 16. Việc tính toán này được mô tả trên Hình A.3. Do đó để xác định đầy đủ ***KS***, chỉ cần mô tả hai lựa chọn hoán vị, cũng giống như lược đồ chuyển dịch sang trái. Một bit trong byte 8-bit của ***KEY*** có thể được khởi động để phát hiện lỗi trong tạo khóa, phân phối và lưu trữ khóa. Các bit 8, 16,..., 64 được sử dụng để bảo đảm là mỗi byte có tính chất lẻ. Lựa chọn hoán vị 1 được xác định bởi Bảng A.6

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Bảng A.6 - Hoán vị khóa PC-1

Bảng trên được chia thành hai phần, phần thứ nhất xác định cách chọn các bit trong $\boldsymbol{C_0}$, phần thứ hai xác định cách chọn các bit trong $\boldsymbol{D_0}$. Các bit của ***KEY*** được đánh số từ 1 đến 64. Các bit trong $\boldsymbol{C_0}$ tương ứng là 57, 49, 41,..., 44 và 36 của ***KEY***, còn các bit trong $\boldsymbol{D_0}$ là các bit 63, 55, 47,..,12 và 4 của ***KEY***. Với $\boldsymbol{C_0}$ và $\boldsymbol{D_0}$ đã xác định, có thể xác định được $\boldsymbol{C_n}$ và $\boldsymbol{D_n}$ từ $\boldsymbol{C_{n-1}}$ và $\boldsymbol{D_{n-1}}$, tương ứng với n = 1, 2,..., 16. Điều này đạt được bằng cách tuân thủ vào lược đồ phép dịch sang trái các khối riêng lẻ sau:

Số lần lặp	Số lần dịch trái
1	1

2	1
3	2
4	2
5	2
6	2
7	2
8	2
Số lần lặp	Số lần dịch trái
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

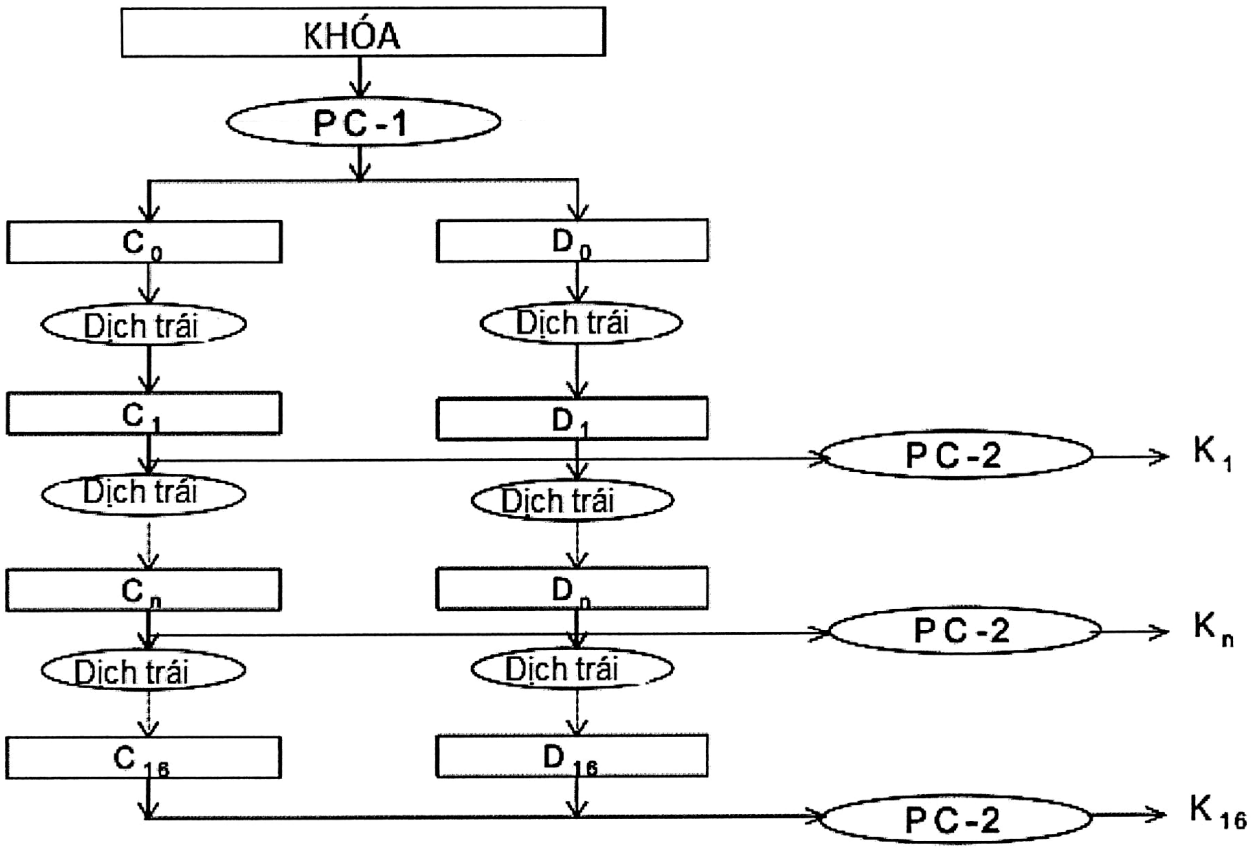
Bảng A.7 – Số các bit khóa được dịch chuyển của mỗi vòng

Ví dụ, C_3 và D_3 thu được từ C_2 và D_2 tương ứng bằng hai dịch chuyển sang trái, và C_{16} và D_{16} thu được từ C_{15} và D_{15} tương ứng bằng một lần dịch sang trái. Trong tất cả các trường hợp, việc dịch một lần sang trái được hiểu là dịch các bit sang trái một vị trí sao cho sau một lần dịch các bit ở 28 vị trí là các bit trước đó ở vị trí 2, 3,..., 28, 1. Việc chọn hoán vị 2 được xác định trên Bảng A.8

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	10	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Bảng A.8 – Hoán vị nén PC-2

Bởi vậy bit thứ nhất của K_n là bit thứ 14 của C_nD_n, bit thứ hai là bit thứ 17,v.v, bit thứ 47 là bit thứ 29, bit thứ 48 là bit thứ 32.



Hình A.3 - Tính toán lược đồ khóa

Phụ lục B

(Quy định)

Các phép biến đổi của AES

B.1. Các phép biến đổi AES

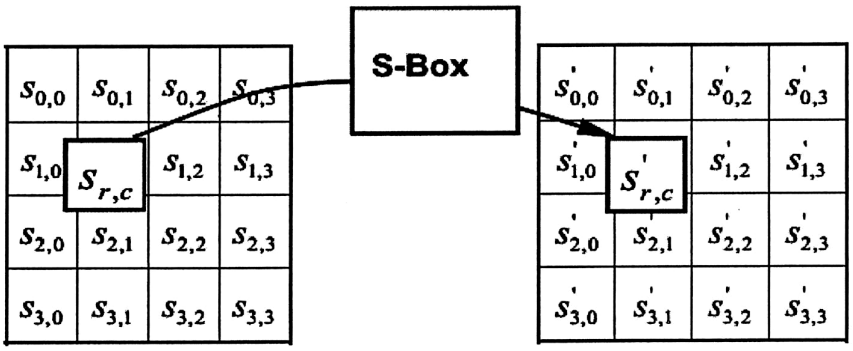
B.1.1. Các phép biến đổi xác định cho AES

Thuật toán AES sử dụng các phép biến đổi $SubBytes()$, $SubBytes^{-1}()$, $ShiftRows()$, $ShiftRows^{-1}()$, $MixColumns()$, $MixColumns^{-1}()$, $AddRoundKey()$, được mô tả dưới đây.

B.1.2. Phép biến đổi $SubBytes()$

Phép biến đổi $SubBytes()$ thực hiện thay thế mỗi byte Trạng thái $s_{i,j}$ bởi giá trị mới $s'_{i,j}$, bằng cách sử dụng bảng thay thế (S-box) khả nghịch.

Hình B.1 minh họa tác động của phép biến đổi SubBytes() lên bảng Trạng thái



Hình B.1: $SubBytes()$ áp dụng S-box cho từng byte của Trạng thái.

S-box được sử dụng trong phép biến đổi $SubBytes()$ và được trình bày theo hệ Hexa trên Bảng B.1

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Bảng B.1: Các S-box của AES

Ví dụ, nếu $s_{1,1}$ = {53} thì giá trị thay thế là giá trị nằm trên giao của hàng có chỉ số '5' và cột có chỉ số '3' của Bảng 5, điều này cho kết quả là $s'_{1,1}$ có giá trị {ed}.

B.1.3. Phép biến đổi $SubBytes^{-1}()$

$SubBytes^{-1}()$ là phép biến đổi nghịch đảo của phép biến đổi $SubBytes()$, trong đó S-box nghịch đảo được áp dụng cho từng byte của Trạng thái. Điều này đạt được bằng cách áp dụng phép biến đổi nghịch đảo được mô tả tại Điều 5.2.4.2

S-box nghịch đảo được sử dụng trong phép biến đổi $SubBytes^{-1}()$ được mô tả trong Bảng B.2.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	sd	9d	84
6	90	d8	ab	00	8c	be	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Bảng B.2: S-box nghịch đảo AES

B.1.4. Phép biến đổi $ShiftRows()$

Trong phép biến đổi $ShiftRows()$, các byte ở ba dòng cuối của Trạng thái được dịch vòng lên một số lượng bytes khác nhau (offsets). Hàng thứ nhất, hàng 0 được giữ nguyên (không dịch chuyển).

Cụ thể, phép biến đổi $ShiftRows()$ được thực hiện như sau:

$S'_{r,c} = S_{r,(c+r)mod4}$ với $0 < r < 4$, và $0 \leq c < 4$, ở đây r là số thứ tự của hàng.

Theo đó các byte dịch sẽ chuyển sang trái (nghĩa là các giá trị thấp hơn của *c* trong một hàng cho trước), trong khi các byte phía trái ngoài cùng dịch vòng sang các vị trí phía phải ngoài cùng của hàng (tức là những giá trị cao hơn của *c* trong hàng cho trước).

Hình B.2 mô tả phép biến đổi $ShiftRows()$, trong đó các byte được dịch vòng sang trái.

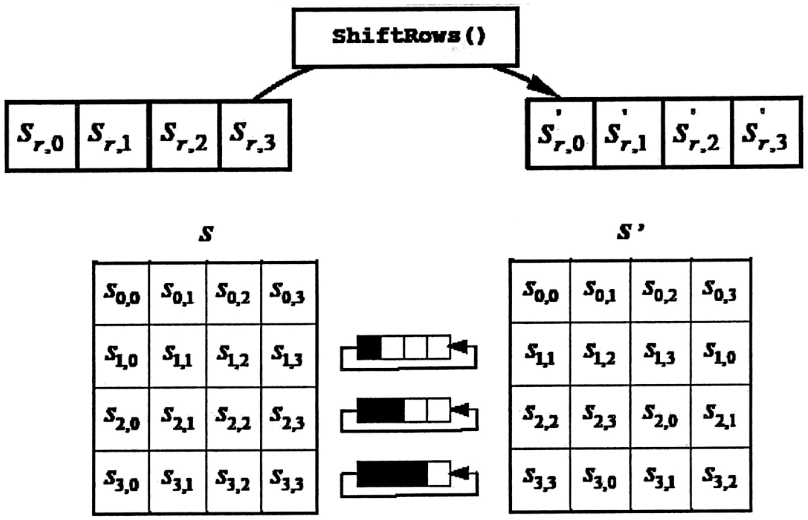
B.1.5. Phép biến đổi $ShiftRows^{-1}()$

$ShiftRows^{-1}()$ là phép nghịch đảo của phép biến đổi $ShiftRows()$. Các bytes trong ba dòng cuối của Trạng thái được dịch vòng lên một số lượng bytes khác nhau. Dòng thứ nhất, dòng 0 không dịch chuyển. Ba dòng dưới cùng được dịch vòng lên $4 - r$ bytes, ở đây r là số thứ tự vòng.

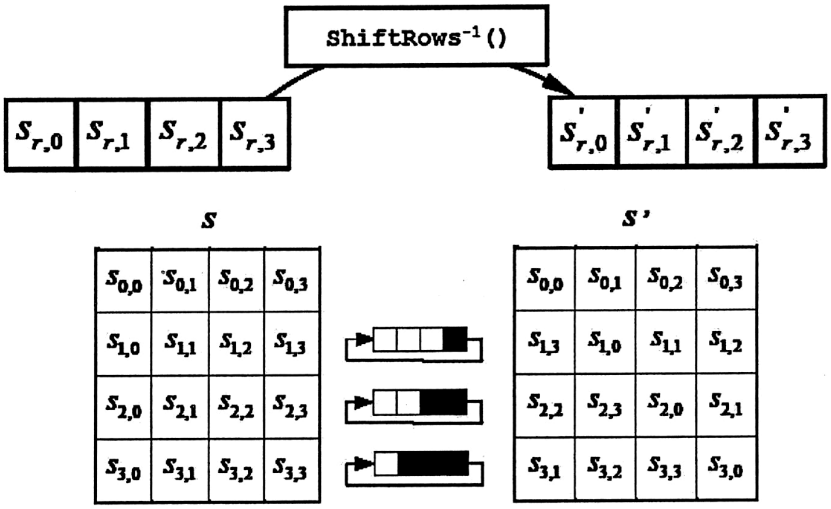
Cụ thể, phép biến đổi $ShiftRows^{-1}()$ được thực hiện như sau:

$S_{r,(c+r)mod4} = S'_{r,c}$ với $0 < r < 4$, và $0 \leq c < 4$.

Hình B.3. mô tả phép biến đổi $ShiftRows^{-1}()$



Hình B.2: $ShiftRows()$ dịch vòng ba dòng cuối của Trạng thái



Hình B.3: $ShiftRows^{-1}()$ dịch vòng ba dòng cuối của Trạng thái.

B.1.6. Phép biến đổi $MixColumns()$

Phép biến đổi $MixColumns()$ thao tác trên Trạng thái, thay mỗi cột bằng cột khác. Các cột của Trạng thái được xem như những đa thức trên trường $GF(2^8)$ và được nhân modulo $x^4 + 1$ với đa thức cố định $a(x)$ cho trước, $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. Phép nhân này có thể viết dưới dạng phép nhân ma trận:

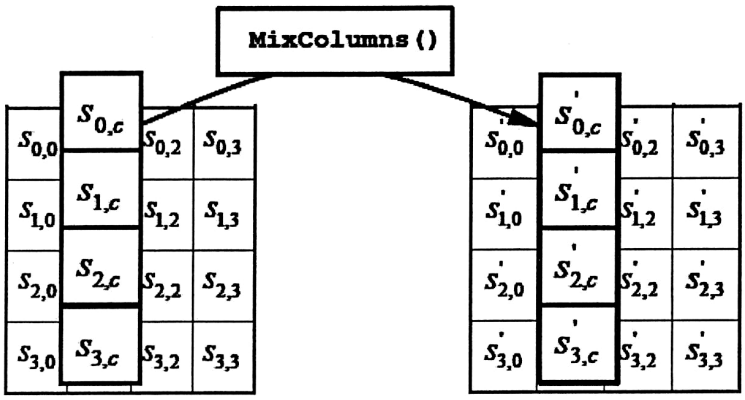
$$s'(x) = a(x) \otimes s(x): \quad \begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{với } 0 \leq c < 4.$$

Kết quả của phép nhân trên là bốn byte trong cột được thay thế như sau:

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$
$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

Toán tử $\oplus$ trong các biểu thức trên biểu thị phép cộng trong trường $GF(2^8)$, với phép cộng bit XOR. Phép nhân được thực hiện theo modulo của đa thức bất khả qui của trường. Trong trường hợp thuật toán AES đó là đa thức $x^8 + x^4 + x^3 + x + 1$.

Hình B.4 mô tả phép biến đổi $MixColumns()$.



Hình B.4 — $MixColumns$ () thao tác trên Trạng thái, thay cột bằng cột khác

B.1.7. Phép biến đổi $MixColumns^{-1}()$

$MixColumns^{-1}()$ là phép biến đổi nghịch đảo của phép biến đổi $MixColumns()$. $MixColumns^{-1}()$ thao tác trên Trạng thái, thay mỗi cột bằng cột khác. Phép biến đổi này có thể biểu diễn dưới dạng phép nhân ma trận, ở đấy mỗi byte được coi như một phần tử của trường hữu hạn $GF(2^8)$:

$$s'(x) = a^{-1}(x) \otimes s(x) \qquad \begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \qquad \text{với } 0 \le c < 4$$

Kết quả của phép nhân là bốn byte trong mỗi cột được thay thế như sau:

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$
$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$
$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

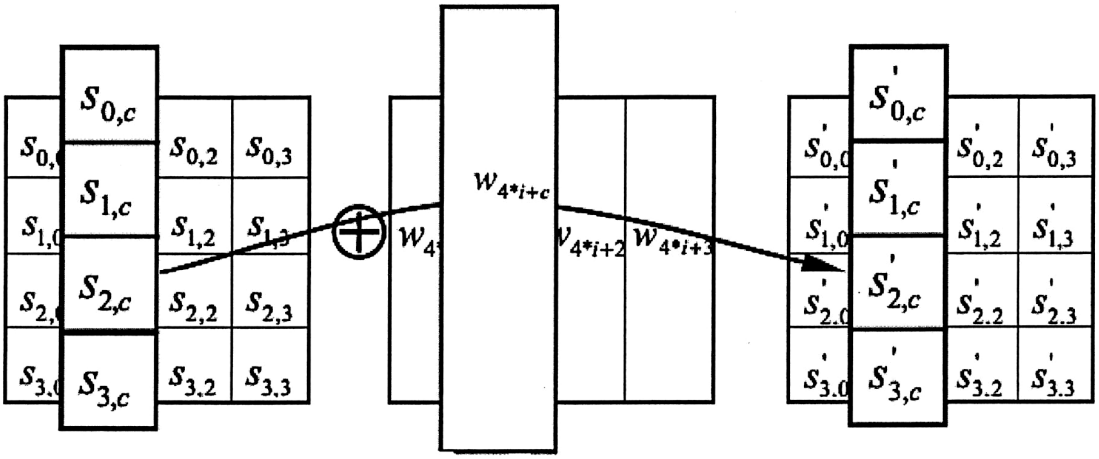
B.1.8. Phép biến đổi $AddRoundKey()$

Trong phép biến đổi $AddRoundKey()$, khóa vòng được cộng vào Trạng thái bằng phép cộng bit đơn giản XOR. Mỗi khóa vòng gồm bốn từ (128 bit) lấy từ lược đồ khóa (được mô tả tại 5.2.5). Bốn từ này được cộng vào cột Trạng thái như sau:

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{(4*i+c)}] \quad \text{với } 0 \le c < 4$$

Ở đây, $0 \le c < 4$ và $w_{(4*i+c)}$ là các từ của lược đồ khóa thứ c của khóa vòng thứ i $W_i = [w_{(4*i)}, w_{(4*i+1)}, w_{(4*i+2)}, w_{(4*i+3)}]$ và i là giá trị thuộc khoảng $0 \le i \le Nr$. Trong phép mã hóa, phép cộng khóa vòng ban đầu xảy ra khi $i = 0$, trước ứng dụng thứ nhất của hàm vòng. Việc áp dụng phép biến đổi $AddRoundKey()$ cho ***Nr*** vòng của phép mã hóa xảy ra khi $1 \le i \le Nr$.

Hoạt động của phép biến đổi $AddRoundKey()$, được minh họa trên Hình B.5. Địa chỉ byte trong các từ của lược đồ khóa được mô tả trong Điều 2.2.2.2.



Hình B.5: $AddRoundKey()$ cộng bit XOR từng cột của Trạng thái với mỗi từ lấy từ lược đồ khóa

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

QCVN 5:2016/BQP

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ CHỮ KÝ SỐ SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on digital signature used in banking

HÀ NỘI - 2016

Mục lục

Lời nói đầu

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

1.2. Đối tượng áp dụng

1.3. Tài liệu viện dẫn

1.4. Giải thích từ ngữ

1.5. Các ký hiệu

2. QUY ĐỊNH KỸ THUẬT

2.1. Chữ ký số

2.1.1. Quy định kỹ thuật

2.1.2. Chữ ký số RSA-PSS

2.1.3. Chữ ký số ECDSA

2.1.4. Chữ ký số DSA

2.1.5. Chữ ký số RSASSA-PKCS1-v1_5

2.2. Hàm băm

3. QUY ĐỊNH VỀ QUẢN LÝ

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

5. TỔ CHỨC THỰC HIỆN

Lời nói đầu

QCVN 5 : 2016/BQP do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ biên soạn, Ban Cơ yếu Chính phủ trình duyệt, Bộ Khoa học và Công nghệ thẩm định và được ban hành theo Thông tư số 161/2016/TT-BQP ngày 21 tháng 10 năm 2016 của Bộ trưởng Bộ Quốc phòng.

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ CHỮ KÝ SỐ SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on digital signature used in banking

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

Quy chuẩn kỹ thuật quốc gia này quy định mức giới hạn của các đặc tính kỹ thuật mật mã của chữ ký số sử dụng trong lĩnh vực ngân hàng.

1.2. Đối tượng áp dụng

Quy chuẩn này áp dụng đối với các doanh nghiệp kinh doanh sản phẩm, dịch vụ mật mã dân sự trong lĩnh vực ngân hàng; các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự.

1.3. Tài liệu viện dẫn

- *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP 800-90A Rev. 1, National Institute of Standards and Technology, June 2015. (*Khuyến cáo cho bộ sinh số ngẫu nhiên sử dụng bộ sinh bit ngẫu nhiên tất định*, NIST SP 800-90A Rev. 1, Viện tiêu chuẩn và công nghệ quốc gia (Mỹ), tháng 6 năm 2015).
- RSA Laboratories. *PKCS#1 v2.1: RSA Cryptography Standard*. June 2002. (Phòng thí nghiệm RSA. *PKCS#1 v2.1: Tiêu chuẩn mật mã RSA*. Tháng 6 năm 2002).
- TCVN 7635:2007 Kỹ thuật mật mã – Chữ ký số.
- NIST, Federal Information Processing Standards Publication 186-4, *Digital Signature Standard (DSS)*, July 2013. (NIST, Tiêu chuẩn xử lý thông tin liên bang, *Chuẩn chữ ký số (DSS)*, tháng 7 năm 2013).
- NIST, Federal Information Processing Standards Publication 180-4, *Secure Hash Standard (SHS)*, August 2015. (NIST, Tiêu chuẩn xử lý thông tin liên bang, Chuẩn *hàm băm an toàn (SHS)*, tháng 8 năm 2015).

1.4. Giải thích từ ngữ

Trong Quy chuẩn này, các từ ngữ dưới đây được hiểu như sau:

1.4.1.

Thông tin không thuộc phạm vi bí mật nhà nước

Là thông tin không thuộc nội dung tin "tuyệt mật", "tối mật" và "mật" được quy định tại Pháp lệnh Bảo vệ bí mật nhà nước ngày 28 tháng 12 năm 2000.

1.4.2.

Mật mã

Là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

1.4.3.

Mật mã dân sự

Là kỹ thuật mật mã và sản phẩm mật mã được sử dụng để bảo mật hoặc xác thực đối với thông tin không thuộc phạm vi bí mật nhà nước.

1.4.4.

Sản phẩm mật mã dân sự

Là các tài liệu, trang thiết bị kỹ thuật và nghiệp vụ mật mã để bảo vệ thông tin không thuộc phạm vi bí mật nhà nước.

1.4.5.

Kỹ thuật mật mã

Là phương pháp, phương tiện có ứng dụng mật mã để bảo vệ thông tin.

1.4.6.

Mã hóa

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là che giấu nội dung thông tin của dữ liệu.

1.4.7.

Giải mã

Phép toán ngược với phép mã hóa tương ứng.

1.4.8.

Mật mã phi đối xứng

Hệ thống dựa trên kỹ thuật mật mã phi đối xứng, trong đó phép biến đổi công khai được sử dụng để mã hóa, phép biến đổi bí mật được sử dụng để giải mã.

1.4.9.

Kỹ thuật mật mã phi đối xứng

Kỹ thuật mật mã phi đối xứng sử dụng hai phép biến đổi liên quan đến nhau, phép biến đổi công khai (được xác định bởi khóa công khai) và phép biến đổi bí mật (được xác định bởi khóa riêng). Cả hai phép biến đổi này có tính chất là cho biết phép biến đổi công khai, về mặt tính toán không thể có khả năng xác định được phép biến đổi bí mật.

1.4.10.

Chữ ký số

Một chuỗi số, kết quả của phép biến đổi mật mã trên thông điệp dữ liệu nhằm cung cấp một phương tiện để kiểm tra tính xác thực của nguồn gốc thông điệp dữ liệu, tính toàn vẹn của dữ liệu và tính không thể chối bỏ của người đã ký.

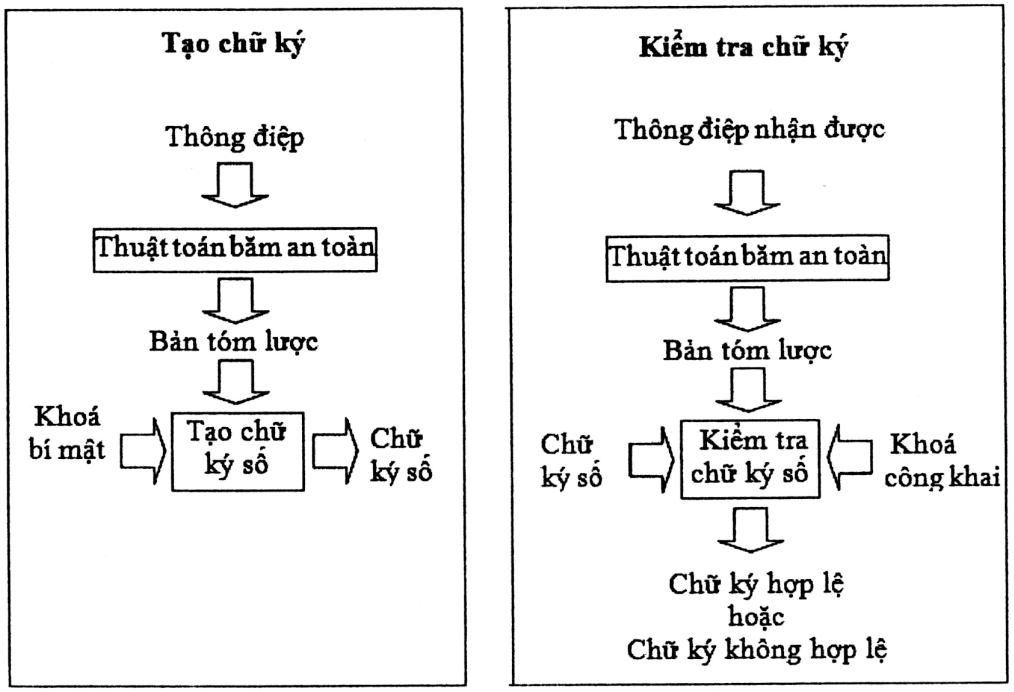
1.5. Các ký hiệu

AES	Tiêu chuẩn mã hóa tiên tiến (*Advanced Encryption Standard*)
$Octet$	Chuỗi bit có độ dài bằng 8
n	Modulo RSA
d	Số mũ bí mật RSA
e	Số mũ công khai RSA
(n, e)	Khóa công khai RSA của người ký
c	Một biểu diễn của bản mã, là số nguyên thuộc $(0, n-1)$
C	Bản mã được biểu diễn ở dạng chuỗi octet
EM	Chuỗi Octet biểu diễn thông điệp đã được ghi mã
$emLen$	Độ dài theo Octet của EM
k	Độ dài modulo n tính theo octet
m	Một biểu diễn của thông điệp (văn bản), là số nguyên thuộc $(0, n-1)$
M	Thông điệp (văn bản), chuỗi octet
$I2OSP$	Hàm cơ sở chuyển đổi từ dạng số nguyên sang chuỗi octet (*Integer-to-Octet-String Primitive*)
$OS2IP$	Hàm cơ sở chuyển đổi từ chuỗi octet sang số nguyên (*Octet-String-to-Integer-Primitive*)
LCM	Bội chung nhỏ nhất (*Least Common Multiplier*)
$nlen$	Độ dài modulo n theo bit
$security_strength$	độ mạnh về an toàn (*security_strength*) là một số nguyên biểu thị lượng tính toán cần thiết để phá hệ mã
N	là ký hiệu độ dài theo bit của số nguyên tố q
L	là ký hiệu độ dài theo bit của số nguyên tố p
$\|\|$	Toán tử nối hai chuỗi
$PKCS$	Tiêu chuẩn mật mã khóa công khai (*Puplic Key Cryptography Standard*) do Phòng thí nghiệm RSA (Mỹ) ban hành.

PSS	Lược đồ ký xác suất (*Probabilistic Signature Scheme*)
RSA	Tên của hệ mã do ba nhà toán học Rivest, Shamir và Adleman sáng tạo ra
$RSAVP$	Phép toán cơ sở phục vụ cho kiểm tra chữ ký RSA
$RSASP$	Phép toán ký RSA cơ sở
$RSASSA$	Lược đồ ký RSA kèm phụ lục (*RSA Signature Scheme with Appendix*)
SHA	Thuật toán băm an toàn (*Secure Hash Algorithm*).
$Word$	Từ (32 bit)
DSA	Thuật toán chữ ký số
EC	Đường cong Elliptic
$ECDSA$	Thuật toán chữ ký số dựa trên đường cong Elliptic

2. QUY ĐỊNH KỸ THUẬT

2.1. Chữ ký số



Hình 1 - Mô tả quá trình tạo và kiểm tra chữ ký số

Chữ ký số là một dạng chữ ký điện tử được tạo ra bằng sự biến đổi một thông điệp dữ liệu sử dụng hệ thống mật mã phi đối xứng theo đó người có được thông điệp dữ liệu ban đầu và khóa công khai của người ký có thể xác định được chính xác:

- Việc biến đổi nêu trên được tạo ra bằng đúng khóa bí mật tương ứng với khóa công khai trong cùng một cặp khóa;
- Sự toàn vẹn nội dung của thông điệp dữ liệu kể từ khi thực hiện việc biến đổi nêu trên.

Quá trình sinh chữ ký số trên một thông điệp dữ liệu yêu cầu sử dụng: 1) hàm băm mật mã thực hiện tính toán trên dữ liệu sẽ được ký, 2) sử dụng khóa mật mã và thuật toán ký để tạo chữ ký số trên đầu ra của hàm băm. Quy chuẩn này quy định sử dụng khóa mật mã trong thuật toán ký số, hàm băm mật mã được sử dụng trong quá trình sinh chữ ký số.

2.1.1. Quy định kỹ thuật

2.1.1.1. Quy định chung:

Quá trình chữ ký số	Sử dụng	Ghi chú
Tạo chữ ký số	Độ an toàn $\geq$ 112 bit: DSA: $\|p\| \geq 2048$ và $\|q\| \geq 224$ RSA: $\|n\| \geq 2048$ EC: $\|n\| \geq 224$	

Xác thực chữ ký số	Độ an toàn ≥ 112 bit: DSA: $\|p\| \geq 2048$ và $\|q\| \geq 224$ RSA: $\|n\| \geq 2048$ EC: $\|n\| \geq 224$	

2.1.1.2. Quy định chi tiết về nguồn ngẫu nhiên:

Các số ngẫu nhiên được sử dụng cho các mục đích khác nhau như để sinh các tham số mật mã, các khóa mật mã, các giá trị ngẫu nhiên dùng một lần và các giá trị thách đố xác thực.

Một số bộ sinh bit ngẫu nhiên tất định DRBG được chấp thuận để sử dụng theo quy định chung bao gồm: HASH_DRBG, HMAC_DRBG và CTR_DRBG.

Các bộ sinh bit ngẫu nhiên RBG tuân theo SP800-90A phiên bản sửa đổi lại năm 2015 để sinh bit ngẫu nhiên cũng được chấp thuận để sử dụng tiếp.

2.1.2. Chữ ký số RSA-PSS

2.1.2.1. Các yêu cầu chung

1. Cặp khóa RSA dùng để ký thì không được dùng cho mục đích khác (chẳng hạn dùng lại để mã thông điệp);
2. Hai số nguyên tố p, q và số mũ bí mật d cần phải được giữ bí mật tránh việc bị truy cập bất hợp pháp, làm lộ hoặc sửa đổi. Modulo n và số mũ công khai e phải được công bố công khai;
3. Mỗi người sử dụng cần có Modulo n riêng;
4. Độ dài của Modulo n ($nlen$) không được nhỏ hơn 2048 bit và nên được thay đổi theo thời gian như sau.

Năm (y)	độ an toàn	$nlen$ tối thiểu
y<2020	112	2048
	128	3072

Trong đó, độ mạnh về an toàn ($security_strength$) là một số nguyên biểu thị lượng tính toán cần thiết để phá hệ mã.

Vì các phương pháp phá hệ mã thường xuyên được hoàn thiện nên cần phải định kỳ 3 đến 5 năm một lần xem xét lại $nlen$ tối thiểu (có thể tham khảo chi tiết yêu cầu này trong tài liệu *NIST Special Publication 800-57: Recommendation for Key Management – Part1: General, January 2016*).

- Phiên bản áp dụng: Áp dụng phiên bản 2.1 của tiêu chuẩn RSA Cryptography Standard PKCS #1 v2.1.

- Áp dụng lược đồ RSAES-OAEP để mã hóa và RSASSA-PSS để ký.

2.1.2.2. Quy định chi tiết về các khóa RSA

1. Số mũ công khai e cần phải được chọn với các ràng buộc sau:
a) Số mũ công khai e cần được chọn trước khi tạo số mũ bí mật d;
b) Số mũ công khai e cần phải là số nguyên dương lẻ sao cho

$$65{,}537 \leq e < 2^{nlen-2security_strength}$$

Với $nlen$ là độ dài của modulo n theo bit.

Chú ý rằng e có thể là giá trị bất kỳ mà thỏa mãn ràng buộc 1(b); p và q sẽ được chọn (trong mục 2) sao cho e là nguyên tố cùng nhau với cả $(p-1)$ và $(q-1)$.

2. Hai số nguyên tố p và q được tạo ngẫu nhiên và giữ bí mật cần phải được chọn với các ràng buộc sau:
a) $(p-1)$ và $(q-1)$ cần phải là nguyên tố cùng nhau với số mũ công khai e;
b) Mỗi một trong bốn số $(p+1)$, $(p-1)$ và $(q+1)$, $(q-1)$ cần phải có các nhân tử nguyên tố lớn hơn $2^{security_strength+20}$;
c) Nhân tử nguyên tố bí mật p, q cần phải được chọn ngẫu nhiên từ các số nguyên tố thoả mãn $(\sqrt{2})(2^{(nlen/2)-1}) \leq q < p \leq (2^{(nlen/2)}-1)$;
d) $|p-q| > 2^{(nlen/2-100)}$.
3. Số mũ bí mật d cần phải được lựa chọn sau khi tạo p và q với các ràng buộc:
a) Số mũ d cần phải lớn hơn $2^{(nlen/2)}$, và
b) $d = e^{-1} mod(LCM((p-1),\ (q-1)))$

(Chi tiết về hàm tạo các tham số RSA có thể tham khảo trong tài liệu *FIPS 186-4: Digital Signature Standard*).

2.1.2.3. Tạo chữ ký số

$RSASSA-PSS-SIGN(K,M)$

Đầu vào:	K	khóa bí mật RSA của người ký
	M	thông điệp sẽ được ký, là một chuỗi octet
Đầu ra:	S	chữ ký, chuỗi octet có độ dài k, với k là độ dài của modulo RSA theo octet
Thông báo lỗi:		"văn bản quá dài", "lỗi định dạng"

Các bước:

1. Mã hóa $EMSA-PSS$: Áp dụng thao tác $EMSA-PSS-ENCODE$ vào văn bản M để tạo ra thông điệp được định dạng EM có độ dài $\lceil (modBits-1)/8 \rceil$ octet sao cho độ dài bit của số nguyên $OS2IP(EM)$ nhiều nhất là $modBits-1$, với $modBits$ là độ dài theo bit của số n (mođun RSA):

$$EM = EMSA-PSS-ENCODE\ (M, modBits-1).$$

Chú ý rằng độ dài octet của EM sẽ bằng $k-1$ nếu $modBits-1$ chia hết cho 8 và bằng k nếu $modBits-1$ không chia hết cho 8. Nếu hàm $EMSA-PSS-ENCODE$ cho ra

thông báo lỗi "văn bản quá dài" thì $RSASSA - PSS - SIGN$ cũng cho ra thông báo lỗi "văn bản quá dài" và dừng lại. Nếu $EMSA - PSS - ENCODE$ cho ra thông báo "lỗi định dạng" thì $RSASSA - PSS - SIGN$ cũng cho ra thông báo "lỗi định dạng" và dừng lại.

2. Chữ ký *RSA*:

a. Chuyển thông điệp đã được định dạng (chuỗi octet) EM thành biểu diễn thông điệp ở dạng số nguyên m.

$$m = OS2IP(EM).$$

b. Áp dụng phép toán cơ sở $RSASP$ với K là khóa bí mật RSA và biểu diễn thông điệp m để tạo ra biểu diễn chữ ký là số nguyên s:

$$s = RSASP(K, m).$$

c. Chuyển chữ ký s dạng số nguyên thành chữ ký S dạng chuỗi octet có độ dài k:

$$S = I2OSP(s, k).$$

Đưa ra chữ ký S.

2.1.2.4. Xác thực chữ ký số

$RSASSA - PSS - VERIFY((n, e), M, S)$

Đầu vào:	(n, e)	khóa công khai RSA của người ký
	M	thông điệp mà chữ ký của nó cần được kiểm tra, là chuỗi octet
	S	chữ ký được kiểm tra, chuỗi octet có độ dài k, với k là độ dài theo octet của số n, modulo RSA
Đầu ra:		"chữ ký hợp lệ" hoặc "chữ ký không hợp lệ"

Các bước:

1. Kiểm tra độ dài: Nếu độ dài của chữ ký S không là k octet, cho ra thông báo lỗi "chữ ký không hợp lệ" và dừng lại;
2. Kiểm tra chữ ký RSA;

a. Chuyển chữ ký S thành biểu diễn chữ ký ở dạng số nguyên s;

$$s = OS2IP(S)$$

b. Áp dụng phép toán cơ sở RSAVP với khóa công khai RSA là (n, e) và biểu diễn chữ ký s để tạo ra m là số nguyên biểu diễn thông điệp;

$$m = RSAVP((n, e), s)$$

c. Chuyển biểu diễn thông điệp m thành thông điệp đã được định dạng EM có độ dài $emLen = \lceil (modBits - 1)/8 \rceil$ octet, với $modBits$ là độ dài theo bit của số n (Modulo RSA):

$$EM = I2OSP\ (m, emLen)$$

Chú ý rằng $emLen$ sẽ bằng $k-1$ nếu $modBits-1$ chia hết cho 8 và bằng k nếu $modBits-1$ không chia hết cho 8. Nếu $I2OSP$ cho ra thông báo lỗi "số nguyên quá lớn" thì $RSASSA-PSS-VERIFY$ cho ra thông báo lỗi "chữ ký không hợp lệ" và dừng lại.

3. *Kiểm tra* $EMSA-PSS$: Áp dụng thao tác kiểm tra $EMSA-PSS-VERIFY$ vào thông điệp M và thông điệp đã được định dạng EM để xác định xem chúng có tương ứng với nhau hay không;

$$Result = EMSA-PSS-VERIFY\ (M, EM, modBits-1).$$

4. Nếu kết quả (Result) là "phù hợp" thì cho ra "chữ ký hợp lệ". Ngược lại sẽ cho ra "chữ ký không hợp lệ".

2.1.3. Chữ ký số ECDSA

2.1.3.1. Quy định chi tiết về các khóa ECDSA:

Kiểm tra tính hợp lệ của các tham số miền $(p, SEED, a, b, G, n, h)$ như sau:

Xâu $SEED$ dùng để sinh ngẫu nhiên đường cong Elliptic xác định trên trường F_p với p là số nguyên tố lẻ.

Trước khi sử dụng một bộ tham số miền, tính hợp lệ của nó phải được kiểm tra theo thuật toán dưới đây:

1. Kiểm tra p là một số nguyên tố lẻ.
2. Kiểm tra a, b, x_G, y_G là các phần tử của trường F_p.
3. Kiểm tra rằng a và b được dẫn xuất tương ứng từ $SEED$.
4. Kiểm tra $(4a^3 + 27b^2)$ khác 0 và $j(E) \neq 0; 1728$ trong F_p
5. Kiểm tra $y_G^2 = x_G^3 + ax_G + b$ trong F_p.
6. Kiểm tra n là nguyên tố và $n > 4\sqrt{p}$.
7. Kiểm tra $nG = O_E$.
8. Kiểm tra đường cong có thuộc danh sách các đường cong yếu:
   a. Thỏa mãn điều kiện MOV, (chú ý rằng một đường cong thỏa mãn điều kiện MOV sẽ không phải là đường cong siêu biến).
   b. Kiểm tra đường cong không kì dị, nghĩa là $\#E \neq p$.

Nếu bất kỳ sự kiểm tra nào ở trên thất bại thì tham số miền phải được xem là không hợp lệ.

Điều kiện MOV được hiểu là không có giá trị k nguyên dương nào $0 < k < B$ để cho $p^k - 1$ chia hết cho n. Trên thực hành hiện nay $|p| = 224$ bit thì người ta xét với $B = 15$ là đủ vì khi đó $|p^k| = 3360 > 2048$.

Các hệ số a, b của đường cong được sinh ngẫu nhiên trên F_p từ đầu vào $SEED$ và có thể kiểm tra được.

Khóa bí mật d phải được sinh ngẫu nhiên trong khoảng $[1, n-1]$.

Đường cong Elliptic xác định trên trường hữu hạn F_p với tối thiểu $|p| = 224$ bit và được xác định cụ thể như sau:

Độ dài bit của n	Độ dài bit của p
224 - 255	|p| = 224
256 - 383	|p| = 256
384 - 511	|p| = 384
≥ 512	|p| = 521

Đại lượng Cofactor được định nghĩa và ký hiệu là $h = \#E(F_p)/n$ tuân theo bảng dưới đây:

Độ dài bit của n	Giá trị h cực đại cho phép
224 - 255	2^{14}
256 - 383	2^{16}
384 - 511	2^{24}
≥ 512	2^{32}

2.1.3.2. Tạo chữ ký số

Thuật toán Chữ ký số Đường cong Elliptic (ECDSA) thực hiện việc sinh chữ ký của thông báo m, làm việc như sau:

Sinh chữ ký $ECDSA$

Đầu vào: Các tham số miền (E, P), khóa bí mật d, thông báo m.

Đầu ra : Chữ ký (r, s).

1. Lấy $0 < k < q$ một cách ngẫu nhiên
2. $(x_R, y_R) \leftarrow kP$
3. $r \leftarrow x_R \bmod q$
4. Nếu $r = 0$ thì chuyển về bước 1
5. $k \leftarrow k^{-1} \bmod q$
6. $e \leftarrow H(m)$
7. $s \leftarrow k(e + rd) \bmod q$
8. Nếu $s = 0$ thì quay về bước 1
9. Trả về (r, s)

2.1.3.3. Xác thực chữ ký số

Để kiểm tra chữ ký số (r, s) của thông báo m, người kiểm tra tính các bước sau ($Q = dP$ - ký hiệu khóa công khai):

Kiểm tra chữ ký $ECDSA$

Đầu vào: Các tham số miền (E, P), khóa công khai Q, thông báo m, chữ ký (r, s)

Đầu ra : Chấp nhận hoặc bác bỏ chữ ký.

1. Kiểm tra rằng $0 < r, s < q$
2. $s' \leftarrow s^{-1} \bmod q$
3. $e \leftarrow H(m)$
4. $h_1 \leftarrow s'e \bmod q$
5. $h_2 \leftarrow s'r \bmod q$
6. $R = (x_R, y_R) \leftarrow h_1 P + h_2 Q$
7. Nếu $R = 0$ thì bác bỏ chữ ký.
8. Nếu $x_R \bmod q = r$ thì chấp nhận, ngược lại thì bác bỏ.

2.1.4. Chữ ký số DSA

2.1.4.1. Quy định chi tiết về các khóa DSA:

Các tham số riêng bí mật k phải là số được sinh ngẫu nhiên $0 < k < q$ với độ dài $|q|$ không nhỏ hơn 224 bit.

2.1.4.2. Tạo chữ ký số

Chọn N là độ dài bit của $q, min(N, outlen)$ biểu thị số nguyên dương nhỏ nhất N và $outlen$ là độ dài bit của khối đầu ra hàm băm.

Chữ ký của một thông điệp M bao gồm cặp số r và s được tính toán như sau:

$r = (g^k \bmod p) \bmod q.$

z = các bit tận cùng bên trái $min(N, outlen)$ của $Hash(M)$.

$s = (k^{-1}(z + xr)) \bmod q.$

Khi tính s, xâu z thu được từ hàm $Hash(M)$ sẽ được biến đổi sang một số nguyên.

Chú ý r có thể được tính bất cứ khi nào nếu biết k, p, q và g. Ví dụ: bất cứ khi nào các tham số miền p, q và g được biết, và k được tính toán trước thì r cũng có thể được tính trước vì thông tin của thông báo được ký không yêu cầu các tính toán của r. Việc tính trước giá trị k, k^{-1} và r sẽ được bảo vệ một cách tương tự như khóa riêng x tới khi tính toán xong s.

Giá trị của r và s sẽ được kiểm tra để xác định xem $r = 0$ hay $s = 0$. Nếu một trong hai $r = 0$ hoặc $s = 0$ thì một giá trị mới của k sẽ được sinh ra và chữ ký sẽ được tính toán lại. Nếu chữ ký số được sinh ra đúng đắn, thì rất hiếm khi xảy ra $r = 0$ hoặc $s = 0$.

Chữ ký (r, s) có thể được truyền đi cùng thông điệp để xác thực.

2.1.4.3. Xác thực chữ ký số

Gọi M', r', và s' là các phiên bản tương ứng đã nhận được của M, r và s; gọi y là khóa công khai của người ký; gọi N là độ dài bit của q, và $\boldsymbol{min}(N, outlen)$ biểu thị số nguyên dương nhỏ nhất N và $outlen$, ở đây $outlen$ là độ dài bit của khối đầu ra hàm băm.

Tiến trình xác thực chữ ký được thực hiện như sau:

1. Kiểm tra nếu một trong hai điều kiện $0 < r' < q$ và $0 < s' < q$ bị vi phạm thì chữ ký bị từ chối và được coi là không hợp lệ.
2. Nếu cả hai điều kiện ở bước 1 được thỏa mãn thì:

   $w = (s')^{-1} mod\ q$

   z = Các bit tận cùng bên trái $min(N, outlen)$ của $Hash(M')$.

   $u1 = (zw)\ mod\ q.$

   $u2 = ((r')w)\ mod\ q.$

   $v = (((g)^{u1}\ (y)^{u2})\ mod\ p)\ mod\ q.$
3. Nếu $v = r'$, khi đó chữ ký được xác thực.
4. Nếu v khác r', khi đó thông báo hoặc chữ ký có thể đã bị thay đổi, có thể do lỗi trong tiến trình sinh chữ ký hoặc chữ ký có thể bị giả mạo.

2.1.5. Chữ ký số RSASSA-PKCS1-v1_5

2.1.5.1. Các yêu cầu chung

1. Cặp khoá RSA dùng để ký thì không được dùng cho mục đích khác (chẳng hạn dùng lại để mã thông điệp);
2. Hai số nguyên tố p, q và số mũ bí mật d cần phải được giữ bí mật tránh việc bị truy cập bất hợp pháp, làm lộ hoặc sửa đổi. Modulo n và số mũ công khai e phải được công bố công khai;
3. Mỗi người sử dụng cần có Modulo n riêng;
4. Độ dài của Modulo n $(nlen)$ không được nhỏ hơn 2048 bit và nên được thay đổi theo thời gian như sau.

Năm (y)	độ an toàn	*nlen* tối thiểu
y<2020	112	2048
	128	3072

Trong đó, độ mạnh về an toàn $(security_strength)$ là một số nguyên biểu thị lượng tính toán cần thiết để phá hệ mã.

Vì các phương pháp phá hệ mã thường xuyên được hoàn thiện nên cần phải định kỳ 3 đến 5 năm một lần xem xét lại $nlen$ tối thiểu (có thể tham khảo chi tiết yêu cầu này trong tài liệu *NIST Special Publication 800-57: Recommendation for Key Management – Part1: General, January 2016*).

2.1.5.2. Quy định chi tiết về các khóa RSA

1. Số mũ công khai e cần phải được chọn với các ràng buộc sau:

a) Số mũ công khai e cần được chọn trước khi tạo số mũ bí mật d;

b) Số mũ công khai e cần phải là số nguyên dương lẻ sao cho

$$65{,}537 \leq e < 2^{nlen-2security_strength}$$

Với $nlen$ là độ dài của modulo n theo bit.

Chú ý rằng e có thể là giá trị bất kỳ mà thỏa mãn ràng buộc 1(b); p và q sẽ được chọn (trong mục 2) sao cho e là nguyên tố cùng nhau với cả $(p-1)$ và $(q-1)$.

2. Hai số nguyên tố p và q được tạo ngẫu nhiên và giữ bí mật cần phải được chọn với các ràng buộc sau:

a) $(p-1)$ và $(q-1)$ cần phải nguyên tố cùng nhau với số mũ công khai e;

b) Mỗi một trong bốn số $(p+1)$, $(p-1)$ và $(q+1)$, $(q-1)$ cần phải có các nhân tử nguyên tố lớn hơn $2^{security_strength+20}$;

c) Nhân tử nguyên tố bí mật p, q cần phải được chọn ngẫu nhiên từ các số nguyên tố thoả mãn $(\sqrt{2})(2^{(nlen/2)-1}) \leq q < p \leq 2^{(nlen/2)} - 1$;

d) $|p-q| > 2^{(nlen/2-100)}$.

3. Số mũ bí mật d cần phải được lựa chọn sau khi tạo p và q với các ràng buộc:

a) Số mũ d cần phải lớn hơn $2^{(nlen/2)}$, và

b) $d = e^{-1} mod\ (LCM((p-1),(q-1)))$

(Chi tiết về hàm tạo các tham số RSA có thể tham khảo trong tài liệu *FIPS 186-4: Digital Signature Standard*).

2.1.5.3. Tạo chữ ký số

$$RSASSA-PKCS1-V1_5-SIGN\ (K, M)$$

Đầu vào:	K	khóa bí mật RSA của người ký
	M	thông báo được ký theo xâu bộ 8
Đầu ra:	S	Chữ ký, một xâu bộ 8 của độ dài k trong đó k là độ dài theo bộ 8 của RSA modulo n
Các lỗi		"Thông báo quá dài"; "Modulo RSA quá ngắn"

Các bước thực hiện:

1. Mã hóa *EMSA-PKCS1-v1_5*:

Áp dụng mã hóa *EMSA-PKCS1-v1_5* đối với thông báo M để tạo ra bản mã EM có độ dài là k bộ 8.

$$EM = EMSA - PKCS1 - V1_5 - ENCODE\ (M, k).$$

2. Chữ ký *RSA*:

a. Biến đổi bản mã EM thành số nguyên m

$$m = OS2IP(EM)\ .$$

b. Áp dụng chữ ký số nguyên thủy $RSASP1$ cho khóa riêng K và thông điệp m để tạo ra số nguyên s

$$s = RSASP1\ (K, m).$$

c. Biến đổi s tương ứng thành chữ ký S có độ dài k bộ 8

$$S = I2OSP\ (s, k).$$

3. Đưa ra chữ ký S.

2.1.5.4. Xác thực chữ ký số

$RSASSA - PKCS1 - V1_5 - VERIFY\ ((n, e), M, S)$

Đầu vào:	(n, e)	khóa công khai RSA của người ký
	M	thông báo (chữ ký số) cần xác thực theo xâu bộ 8
	S	Chữ ký cần xác thực là xâu bộ 8 độ dài k trong đó k là độ dài theo bộ 8 của RSA modulo n
Đầu ra:		"Chữ ký hợp lệ" hoặc "chữ ký không hợp lệ"
Các lỗi:		"Thông điệp quá dài"; "Modulo RSA quá ngắn"

Các bước thực hiện:

1. Kiểm tra độ dài: Nếu độ dài của chữ ký S không phải là k các bộ 8 thì xuất đầu ra "chữ ký không hợp lệ" và dừng lại.

2. Xác thực *RSA*:

a. Biến đổi S thành số nguyên s

$$s = OS2IP\ (S).$$

b. Sử dụng $RSAVP1$ nguyên thủy cho khóa RSA công khai (n, e) và s để tạo ra số nguyên m

$$m = RSAVP1\ ((n, e), s).$$

Nếu đầu ra $RSAVP$ "chữ ký số đại diện nằm ngoài dải hợp lệ" thì đưa ra "chữ ký số không hợp lệ" và dừng lại.

c. Biến đổi m thành EM có độ dài k các bộ 8:

$$EM = I2OSP\ (m, k).$$

Nếu đầu ra $I2OSP$ "số nguyên quá lớn" thì đưa ra "chữ ký số không hợp lệ" và dừng lại.

3. *EMSA-PKCS1-v1_5 encoding*:

Áp dụng mã hóa *EMSA-PKCS1-v1_5* đối với thông điệp M để tạo ra bản mã thứ 2 EM' có độ dài k các bộ 8:

$$EM' = EMSA - PKCS1 - V1_5 - ENCODE\ (M, k).$$

4. So sánh EM *và* EM' nếu giống nhau thì chữ ký hợp lệ, ngược lại chữ ký là không hợp lệ.

2.2. Hàm băm

Hàm băm: Một thuật toán chuyển đổi mỗi thông báo biểu diễn dưới dạng bit có độ dài bất kỳ thành một chuỗi bit có độ dài cố định. Chuỗi bit có độ dài cố định đó được gọi là "giá trị băm", "mã băm", hai đơn giản là "tóm lược" của thông báo đầu vào. Các thuật toán băm mật mã được thiết kế sao cho thỏa mãn các tính chất sau:

- Tính một chiều hay tính kháng tiền ảnh: Không thể tìm được trong thời gian cho phép một thông báo có giá trị băm cho trước;
- Tính kháng tiền ảnh thứ hai: Cho trước thông điệp M1, không thể tìm được trong thời gian cho phép một thông điệp M2 khác M1 sao cho giá trị băm của M1 và M2 là như nhau;
- Tính kháng xung đột: Không thể tìm được hai chuỗi bit khác nhau có cùng một giá trị băm.

Trong Quy chuẩn này các hàm băm sau đây được phép sử dụng.

Độ an toàn	Hàm băm áp dụng
128	SHA-256, SHA-512/256, SHA3-256
192	SHA-384, SHA3-384
≥ 256	SHA-512, SHA3-512

3. QUY ĐỊNH VỀ QUẢN LÝ

3.1. Các mức giới hạn của đặc tính kỹ thuật mật mã và yêu cầu quản lý của chữ ký số nêu tại Quy chuẩn này là các chỉ tiêu chất lượng phục vụ được quản lý theo quy định về quản lý chất lượng sản phẩm, dịch vụ mật mã dân sự được quy định tại Luật An toàn thông tin mạng ngày 19 tháng 11 năm 2015.

3.2. Hoạt động kiểm tra chất lượng sản phẩm, dịch vụ mật mã được cơ quan quản lý nhà nước có thẩm quyền tiến hành định kỳ hàng năm hoặc đột xuất.

3.3. Một số quy định về ngưỡng thời gian và độ an toàn khóa cụ thể:

- Quy định nghiệp vụ chung đối với độ an toàn khóa mật mã:

Độ an toàn theo bit	Thời hạn sử dụng quy định
96	Đến cuối năm 2020
112	Đến cuối năm 2030
≥ 128	Từ năm 2030

- Thời hạn sử dụng được quy định chi tiết đối với độ an toàn khóa mật mã khóa công khai tính theo bit:

Thời hạn Quy định	RSA	DSA	ECDSA (Độ dài p)
2020	$\|p\| = 1536$	$\|p\| = 1536, \|q\| = 192$	192 - 224
2030	$\|p\| = 2048$	$\|p\| = 2048, \|q\| = 224$	224 - 255
Sau 2030	$\|p\| = 3072$	$\|p\| = 3072, \|q\| = 256$	256

- Các hàm băm tương ứng theo Độ an toàn bit:

Hàm băm	Độ an toàn
SHA-224	112
SHA-256	128
SHA-512	256
SHA-384	192
WHIRLPOOL	256

- Độ an toàn theo bit quy đổi giữa RSA, DSA và ECDSA như sau:

Độ an toàn	ECDSA	RSA	DSA
112	224	2048	2048
128	256	3072	3072
192	384	7680	7680
256	512	15360	15360

3.4. Quy định về an toàn cài đặt và sử dụng:

Các thuật toán chữ ký số khi cài đặt phần mềm và phần cứng còn cần có đủ khả năng chống lại các tấn công kênh kề nhất là chống lại việc tính ra được các bit khóa trong quá trình thực hiện thuật toán.

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

4.1. Các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự có trách nhiệm đảm bảo tuân thủ Quy chuẩn này và chịu sự kiểm tra của cơ quan quản lý nhà nước theo quy định.

4.2. Doanh nghiệp cung cấp sản phẩm, dịch vụ mật mã dân sự cho các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) có trách nhiệm thực hiện công bố hợp quy sản phẩm, dịch vụ mật mã dân sự phù hợp với Quy chuẩn này. Việc công bố hợp quy thực hiện theo Thông tư số 28/2012/TT-BKHCN ngày 12 tháng 12 năm 2012 của Bộ Khoa học và Công nghệ.

4.3. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ có trách nhiệm tiếp nhận đăng ký công bố hợp quy, thực hiện quản lý, hướng dẫn và kiểm tra việc công bố hợp quy.

5. TỔ CHỨC THỰC HIỆN

5.1. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ có trách nhiệm hướng dẫn, tổ chức triển khai quản lý kỹ thuật mật mã của Chữ ký số theo Quy chuẩn này.

5.2. Trong trường hợp các quy định nêu tại Quy chuẩn kỹ thuật quốc gia này có sự thay đổi, bổ sung hoặc được thay thế thì thực hiện theo quy định tại văn bản mới./.

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

QCVN 6:2016/BQP

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ QUẢN LÝ KHÓA SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on key management used in banking

HÀ NỘI – 2016

MỤC LỤC

Lời nói đầu

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

1.2. Đối tượng áp dụng

1.3. Tài liệu viện dẫn

1.4. Giải thích từ ngữ

1.5. Các ký hiệu

2. QUY ĐỊNH KỸ THUẬT

2.1. Các yêu cầu đối với giao thức thỏa thuận và vận chuyển khóa

2.2. Giao thức thỏa thuận khóa sử dụng kỹ thuật mật mã phi đối xứng trên trường hữu hạn

2.3. Giao thức thỏa thuận khóa sử dụng mật mã trên đường cong elliptic

2.3.1. Giao thức DH

2.3.2. Thỏa thuận khóa MQV

2.4. Hàm dẫn xuất khóa KDF

2.4.1. Hàm dẫn khóa 1

2.4.2. Hàm dẫn xuất khóa 2

2.5. Giao thức vận chuyển khóa bí mật

2.6. Giao thức vận chuyển khóa công khai

2.6.1 Giao thức vận chuyển khóa công khai không sử dụng bên thứ ba tin cậy

2.6.2 Giao thức vận chuyển khóa công khai sử dụng bên thứ ba tin cậy

2.7. Quy định kỹ thuật cho các tham số

2.7.1. Quy định về nguồn ngẫu nhiên

2.7.2. Quy định đối với tham số RSA

2.7.3. Hệ mật dựa trên Logarit rời rạc DL

2.7.4. Hệ mật ECC

2.7.5. Độ an toàn theo bit quy đổi giữa RSA, DL và ECC

3. QUY ĐỊNH VỀ QUẢN LÝ

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

5. TỔ CHỨC THỰC HIỆN

Lời nói đầu

QCVN 6 : 2016/BQP do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ biên soạn, Ban Cơ yếu Chính phủ trình duyệt, Bộ Khoa học và Công nghệ thẩm định và được ban hành theo Thông tư số 161/2016/TT-BQP ngày 21 tháng 10 năm 2016 của Bộ trưởng Bộ Quốc phòng.

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ QUẢN LÝ KHÓA SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on key management used in banking

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

Quy chuẩn kỹ thuật quốc gia này quy định các yêu cầu về quản lý khóa mật mã sử dụng kỹ thuật mật mã phi đối xứng để bảo mật dữ liệu trong lĩnh vực ngân hàng bao gồm: Sinh khóa bí mật dùng để liên lạc giữa hai thực thể bằng cơ chế thỏa thuận khóa và dẫn xuất khóa, sử dụng kỹ thuật mật mã phi đối xứng; sinh khóa bí mật cho một thực thể bởi thực thể khác bằng cơ chế truyền khóa sử dụng mật mã phi đối xứng; vận chuyển khóa công khai của một thực thể đến một thực thể khác bằng đường truyền có bảo vệ; quy định về quản lý sử dụng khóa mật mã an toàn.

Quy định về tạo khóa, đăng ký khóa, thu hồi, cài đặt, khôi phục và các vấn đề thuộc quản lý khóa khác không thuộc phạm vi điều chỉnh của quy chuẩn này.

1.2. Đối tượng áp dụng

Quy chuẩn này áp dụng đối với các doanh nghiệp kinh doanh sản phẩm, dịch vụ mật mã dân sự trong lĩnh vực ngân hàng; các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự.

1.3. Tài liệu viện dẫn

- TCVN 7817-3:2007 (ISO/IEC 11770-3:1999) Công nghệ thông tin – Kỹ thuật mật mã – Quản lý khóa – Phần 3: Các cơ chế sử dụng kỹ thuật phi đối xứng.
- ISO/IEC 11770-3:2015 Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques.
- Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, SP 800-56A Revision 2, National Institute of Standards and Technology, May 2013.
- *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP 800-90A Rev. 1, National Institute of Standards and Technology, June 2015. (*Khuyến cáo cho bộ sinh số ngẫu nhiên sử dụng bộ sinh bit ngẫu nhiên tất định*, NIST SP 800-90A Rev. 1, Viện tiêu chuẩn và công nghệ quốc gia (Mỹ), tháng 6 năm 2015).

1.4. Giải thích từ ngữ

Trong Quy chuẩn này, các từ ngữ dưới đây được hiểu như sau:

1.4.1.

Thông tin không thuộc phạm vi bí mật nhà nước

Là thông tin không thuộc nội dung tin "tuyệt mật", "tối mật" và "mật" được quy định tại Pháp lệnh Bảo vệ bí mật nhà nước ngày 28 tháng 12 năm 2000.

1.4.2.

Mật mã

Là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

1.4.3.

Mật mã dân sự

Là kỹ thuật mật mã và sản phẩm mật mã được sử dụng để bảo mật hoặc xác thực đối với thông tin không thuộc phạm vi bí mật nhà nước.

1.4.4.

Sản phẩm mật mã dân sự

Là các tài liệu, trang thiết bị kỹ thuật và nghiệp vụ mật mã để bảo vệ thông tin không thuộc phạm vi bí mật nhà nước.

1.4.5.

Kỹ thuật mật mã

Là phương pháp, phương tiện có ứng dụng mật mã để bảo vệ thông tin.

1.4.6.

Mã hóa

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là giấu nội dung thông tin của dữ liệu.

1.4.7.

Giải mã

Phép toán ngược với phép mã hóa tương ứng.

1.4.8.

Mã phi đối xứng

Hệ thống dựa trên kỹ thuật mật mã phi đối xứng, trong đó phép biến đổi công khai được sử dụng để mã hóa, phép biến đổi bí mật được sử dụng để giải mã.

1.4.9.

Kỹ thuật mật mã phi đối xứng

Kỹ thuật mật mã phi đối xứng sử dụng hai phép biến đổi liên quan đến nhau, phép biến đổi công khai (được xác định bởi khóa công khai) và phép biến đổi bí mật (được xác định bởi khóa riêng). Cả hai phép biến đổi này có tính chất là cho biết phép biến đổi công khai, về mặt tính toán không thể có khả năng xác định được phép biến đổi bí mật.

1.4.10.

Thẻ khóa

Thông điệp quản lý khóa được gửi từ một thực thể tới một thực thể khác trong quá trình thực hiện một cơ chế quản lý khóa.

1.4.11.

Vận chuyển khóa

Tiến trình truyền một khóa từ một thực thể đến một thực thể khác với bảo vệ thích hợp.

1.4.12.

Xác thực thực thể lẫn nhau

Sự xác thực giữa hai thực thể đảm bảo về định danh của mỗi thực thể.

1.4.13.

Xác thực khóa từ thực thể *A* đến thực thể *B*

Đảm bảo cho B rằng chỉ có *A* là thực thể sở hữu khóa đúng.

1.4.14.

Xác thực khóa hai chiều

Đảm bảo xác thực khóa từ *A* đến *B* và từ *B* đến *A*.

1.4.15.

Xác nhận khóa từ *A* đến *B*

Đảm bảo cho thực thể *B* là thực thể *A* sở hữu khóa đúng.

1.4.16.

Xác nhận khóa hai chiều

Đảm bảo xác nhận khóa từ *A* đến *B* và từ *B* đến *A*.

1.4.17.

Thiết lập khóa

Quá trình đảm bảo sự khả dụng một khóa bí mật dùng chung cho một hoặc nhiều thực thể. Thiết lập khóa bao gồm thỏa thuận khóa và vận chuyển khóa.

1.4.18.

Thỏa thuận khóa

Tiến trình kiến tạo một khóa bí mật dùng chung giữa hai thực thể theo cách mà không có bên nào có thể định trước giá trị cho khóa.

1.5. Các ký hiệu

ID_A, ID_B	Định danh của các thực thể A và B
$Cert_X$	Chứng chỉ khóa công khai của thực thể X
$F(h, g)$	Hàm thỏa thuận khóa
$HASH$	Hàm băm
DH	Diffie-Hellman

MQV	Menezes-Qu-vanstone
K	Khóa bí mật cho hệ mật đối xứng
K_{AB}	Khóa bí mật chia sẻ giữa hai thực thể A và B
KT_{Ai}	Thông báo thỏa thuận khóa được gửi bởi thực thể A sau giai đoạn xử lý i
r	Số ngẫu nhiên được sinh trong quá trình thực hiện của một lược đồ.
H	Tập các phần tử có thể của r
S_X	Hàm tạo chữ ký sử dụng khóa riêng của thực thể X
V_X	Hàm kiểm tra chữ ký của thực thể X
$\|\|$	Phép nối hai phần tử dữ liệu với nhau
$\lceil x \rceil$	Số nguyên nhỏ nhất lớn hơn hoặc bằng số thực x
$len(x)$	Độ dài số nguyên x tính bằng bit
$nlen$	Độ dài modulo n tính theo bit
(n, e)	Khóa công khai theo RSA
(n, d)	Khóa riêng theo RSA
L	Độ dài số nguyên tố p trong bài toán logarit rời rạc
N	Độ dài số nguyên tố q trong bài toán logarit rời rạc
q	Ước nguyên tố của $p - 1$
$\#(E)$	Cấp hay lực lượng của đường cong elliptic E
m	Bậc của điểm sinh G
$X(P)$	Hoành độ x của điểm P trên đường cong elliptic E
h	Các đồng thừa số được tính theo công thức $h = \#E/m$
l	Thừa số phụ trong phép nhân đồng thừa số $l = h^{-1} mod\ m$
$\pi(P)$	Phép biến đổi điểm P trên đường cong E thành số nguyên $\pi(P) = \left(X(P) mod 2^{\lceil \rho/2 \rceil}\right) + 2^{\lceil \rho/2 \rceil}, \rho = \lceil log_2 n \rceil$

2. QUY ĐỊNH KỸ THUẬT

2.1. Các yêu cầu đối với giao thức thỏa thuận và vận chuyển khóa

Điều này quy định các giao thức thiết lập khóa bí mật dùng để mã dữ liệu được trao đổi giữa hai thực thể A và B, vận chuyển khóa bí mật từ thực thể A sang thực thể B, vận chuyển khóa công khai của thực thể A sang thực thể B.

Để thực hiện giao thức, mỗi thực thể X đảm bảo các điều kiện sau:

- Sở hữu cặp khóa để ký và kiểm tra chữ ký (S_X, V_X) được cơ quan thẩm quyền cấp dưới dạng chứng thư số $Cert_X$.
- Sử dụng một cặp khóa công khai để mã hóa (E_X, D_X) được quy định tại Điều 2.7 của Quy chuẩn này.
- Sử dụng chung với thực thể thứ hai hàm thỏa thuận khóa F là một trong hai hàm được xác định tại Điều 2.2.2, một hàm dẫn xuất khóa KDF được quy định tại Điều 2.4 của Quy chuẩn này và một hàm kiểm tra mật mã $MAC_{K_{AB}}$ dưới dạng hàm băm được quy định tại Điều 2.2 của QCVN 5 : 2016/BQP Quy chuẩn kỹ thuật quốc gia về Chữ ký số sử dụng trong lĩnh vực ngân hàng.
- Mỗi thực thể được tiếp cận các khóa công khai của thực thể kia theo cơ chế vận chuyển khóa công khai tại Điều 2.6.

2.2. Giao thức thỏa thuận khóa sử dụng kỹ thuật mật mã phi đối xứng trên trường hữu hạn

Thỏa thuận khóa bí mật giữa hai thực thể A và B được thực hiện qua 5 bước:

Bước 1-4. Hai bên thỏa thuận bí mật chia sẻ K_{AB}.

Bước 5. Hai bên sử dụng hàm dẫn xuất khóa KDF được quy định tại Điều 2.4 để thiết lập khóa bí mật chung *K*.

Lược đồ thỏa thuận khóa bí mật chia sẻ K_{AB} được thể hiện trên Hình 1.

2.2.1 Giao thức

Bước 1. Kiến thiết thẻ khóa (A1): Thực thể A sinh một giá trị ngẫu nhiên và bí mật r_A thuộc H, tính $F(r_A, g)$, kiến thiết thẻ khóa KT_{A1} và gửi tới thực thể B:

$$KT_{A1} = F(r_A, g) \parallel Text1$$

Bước 2. Xử lý thẻ khóa và Kiến thiết khóa (B1): Thực thể B sinh một giá trị ngẫu nhiên và bí mật r_B thuộc H, tiếp đó tính $F(r_B, g)$, rồi tính khóa bí mật dùng chung:

$$K_{AB} = F(r_B, F(r_A, g))$$

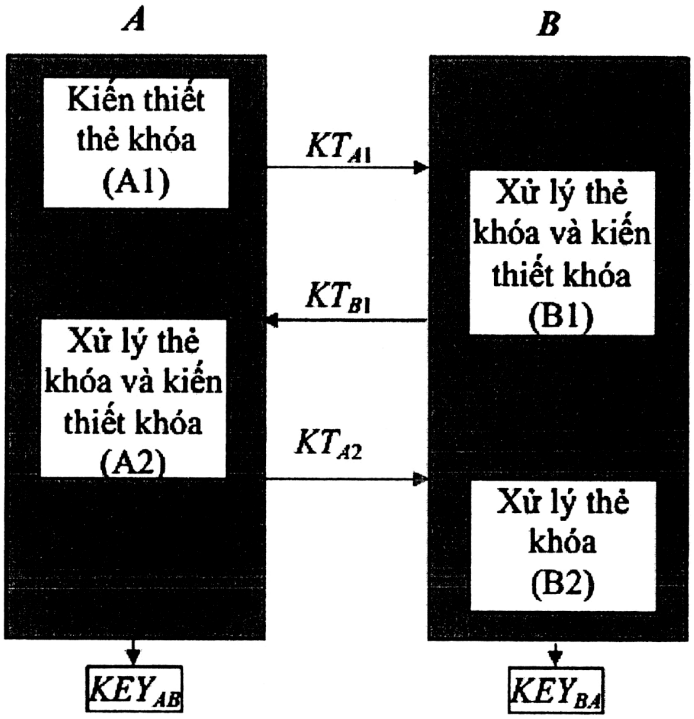
Tiếp theo, thực thể *B* tạo thẻ khóa KT_{B1}được ký như sau:

$$KT_{B1} = S_B(DB_1) \parallel MAC_{K_{AB}}(DB_1) \parallel Text3$$

trong đó

$$DB_1 = F(r_B, g) \parallel F(r_A, g) \parallel ID_A \parallel Text2$$

và gửi ngược trở lại cho thực thể A.



Hình 1: Giao thức thỏa thuận khóa bí mật chia sẻ K_{AB}

Bước 3. Xử lý thẻ khóa (A2): Thực thể A kiểm tra chữ ký của thực thể B trên thẻ khóa KT_{B1} bằng cách sử dụng khóa kiểm tra công khai của B, kiểm tra định danh phân biệt của A và giá trị $F(r_A, g)$ đã được gửi ở Bước 1 **(A1)**. Nếu quá trình kiểm tra thành công thì thực thể A sẽ tính khóa bí mật dùng chung là:

$$K_{AB} = F(r_A, F(r_B, g))$$

Thực thể A lại sử dụng khóa bí mật dùng chung K_{AB} để kiểm tra giá trị kiểm tra mật mã $MAC_{K_{AB}}(DB_1)$. Sau đó thực thể A tạo thẻ khóa KT_{A2} được ký như sau:

$$KT_{A2} = S_A(DB_2) \parallel MAC_{K_{AB}}(DB_2) \parallel Text5$$

trong đó

$$DB_2 = F(r_A, g) \parallel F(r_B, g) \parallel ID_B \parallel Text4$$

và gửi thẻ này tới thực thể B.

Bước 4. Xử lý thẻ khóa (B2): Thực thể B kiểm tra chữ ký của thực thể A trên thẻ khóa KT_{A2} sử dụng khóa kiểm tra công khai của A, sau đó kiểm tra định danh phân biệt của B và kiểm tra các giá trị $F(r_A, g)$,và $F(r_B, g)$, xem có phù hợp với các giá trị được trao đổi ở các bước trước hay không. Nếu quá trình kiểm tra thành công thì thực thể B sẽ kiểm tra giá trị kiểm tra mật mã $MAC_{K_{AB}}(DB_2)$ bằng cách tính:

$$K_{AB} = F(r_A, F(r_B, g))$$

Bước 5. Thực thể A và thực thể B tính khóa bí mật chung K

Các thực thể A và B sử dụng hàm dẫn xuất khóa tại Điều 2.4 để tính khóa

$$K = KDF(K_{AB}, OtherInput)$$

Trong đó $OtherInput$ chứa các định danh ID_A, ID_B của A,B và các thông tin khác do A và B thỏa thuận.

CHÚ THÍCH: Các trường $Text1$, $Text2$, $Text3$, $Text4$, $Text5$ là những trường dữ liệu chứa chứng thư số của A và B, và có thể một số thông tin khác như tem thời gian, định danh phiên liên lạc, v.v..)

2.2.2. Hàm thỏa thuận khóa

2.2.2.1. Hàm thỏa thuận khóa DH

Cho trường hữu hạn nguyên tố F_p, ký hiệu tập $H = \{1, \dots, p-2\}$; g là phần tử sinh thuộc hàm thỏa thuận khóa Diffie-Hellman (DH) được xác định theo công thức:

$$F(h, g) = g^h, h \in H$$

CHÚ THÍCH: Với hàm thỏa thuận DH thì thẻ khóa KT_{A1} có dạng:

$$KT_{A1} = g^{r_A} \parallel Text_A$$

2.2.2.2. Hàm thỏa thuận khóa MQV

- $(a, A); (b, B)$ là cặp khóa phi đối xứng tĩnh của A và B, $A = g^a, B = g^b$
- $(x, X), (y, Y)$ là cặp khóa phi đối xứng tức thời của A và B, $X = g^x, Y = g^y$, $d = 2^l + (X \bmod 2^l), e = 2^l + (Y \bmod 2^l), l = |p|/2$, l độ dài p tính bằng bit, $F(x, g) = (YB^e)^{x+da} = (XA^d)^{y+eb}$

2.3. Giao thức thỏa thuận khóa sử dụng mật mã trên đường cong elliptic

2.3.1. Giao thức DH

Bước 1. Kiến thiết thẻ khóa (A1)

Thực thể A chọn ngẫu nhiên và bí mật số r_A thuộc khoảng $\{2, \dots, m-2\}$, tính $r_A G$, kiến thiết thẻ khóa $KT_{A1} = r_A G$ và gửi cho B.

Bước 2. Xử lý thẻ khóa và kiến thiết thẻ khóa (B1)

Thực thể B kiểm tra thẻ khóa KT_{A1} có phải là điểm nằm trên đường cong Elliptic hay không (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1). Thực thể B chọn ngẫu nhiên và bí mật r_B thuộc khoảng $\{2, \dots, m-2\}$, tính $r_B G$, tính bí mật chia sẻ $K_{AB} = (r_B l)(hKT_{A1})$, kiến thiết thẻ được ký hiệu KT_{B1}:

$KT_{B1} = S_B(DB_1) \parallel MAC_{K_{AB}}(DB_1)$ với $DB_1 = r_B G \parallel KT_{A1} \parallel ID_A \parallel Text1$

và gửi cho A.

Bước 3. Xử lý thẻ khóa (A2)

Thực thể A kiểm tra chữ ký của B trên thẻ khóa KT_{B1} sử dụng khóa kiểm tra công khai của B. Nếu sử dụng lược đồ chữ ký có khôi phục bản rõ thì việc kiểm tra bao gồm cả việc khôi phục khối dữ liệu DB_1 từ chữ ký và kiểm tra liệu định danh phân biệt của A và giá trị $r_A G$ có chứa trong DB_1 không. Nếu sử dụng chữ ký có đính kèm bản rõ thì việc kiểm tra bao gồm cả việc thiết kế lại khối dữ liệu DB_1 sử dụng giá trị trong KT_{A1}, định danh phân biệt của A, giá trị nhận được $r_B G$ và kiểm tra chữ ký trên khối dữ liệu này.

Tiếp đó thực thể A kiểm tra, liệu giá trị $r_B G$ nhận được từ KT_{B1} có phải là điểm trên đường cong elliptic hay không (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1). Nếu đúng thì A tính khóa chia sẻ $K_{AB} = (r_A.l)(h.r_B G)$.

Sử dụng K_{AB}, thực thể A kiểm tra $MAC_{K_{AB}}(DB_1)$. Tiếp đó A thiết kế thẻ khóa có ký hiệu $KT_{A2} = S_A(DB_2) \parallel MAC_{K_{AB}}(DB_2)$ ở đây $DB_2 = r_A G \parallel r_B G \parallel B \parallel Text2$ ($DB_2 = r_A G \parallel r_B G \parallel ID_B \parallel Text2$) và gửi cho B.

Bước 4. Xử lý thẻ khóa (B2)

Thực thể B kiểm tra chữ ký của A trên thẻ khóa KT_{A2} bằng cách sử dụng khóa kiểm tra công khai của A. Nếu sử dụng sơ đồ chữ ký có khôi phục bản rõ thì điều này bao gồm cả việc khôi phục khối dữ liệu DB_2 từ chữ ký và kiểm tra định danh phân biệt của B, các giá trị $r_A G$ và $r_B G$ có chứa trong khối này không. Nếu sử dụng chữ ký có đính kèm bản rõ thì việc kiểm tra bao gồm cả việc thiết kế lại khối dữ liệu DB_2 sử dụng các giá trị trong KT_{A1}và KT_{B1}, định danh phân biệt của B và kiểm tra chữ ký trên khối dữ liệu này.

Nếu việc kiểm tra thành công thì thực thể B kiểm tra $MAC_{K_{AB}}(DB_2)$ sử dụng khóa chia sẻ $K_{AB} = (r_B l)(hKT_{A1})$.

Bước 5. Thiết lập khóa bí mật K

Thực thể A và B sử dụng hàm dẫn xuất khóa KDF tại Điều 2.4 để thiết lập khóa bí mật.

$$K = KDF(K_{AB}, OtherInput).$$

2.3.2. Thỏa thuận khóa MQV

Bước 1. Thiết kế thẻ khóa (A1)

Thực thể A chọn ngẫu nhiên và bí mật số r_A thuộc khoảng $\{2,...,m-2\}$, tính $r_A G$ và kiến thiết thẻ khóa $KT_{A1} = r_A G$ và gửi cho thực thể B.

Bước 2. Kiến thiết thẻ khóa (B1)

Thực thể B kiểm tra thẻ KT_{A1} có phải là điểm trên đường cong elliptic (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1:2016). Thực thể B chọn ngẫu nhiên và bí mật số r_B thuộc khoảng $\{2,..., m-2\}$, tính $r_B G$, thiết kế thẻ khóa $KT_{B1} = r_B G$, tiếp đó tính khóa bí mật chia sẻ K_{AB}:

$$K_{AB} = ((r_B + \pi(KT_{B1})d_B).l)(h.(KT_{A1} + \pi(KT_{A1})P_A))$$

Tiếp đó B tính $K = KDF(K_{AB})$ và gửi $f_{K_{AB}}(2, KT_{A1}, KT_{B1})$ gửi $MAC_K(2, KT_{A1}, KT_{B1})$ cho A cùng với thẻ khóa KT_{B1}.

Bước 3. Kiến thiết thẻ khóa (A2)

Thực thể A tính khóa bí mật chia sẻ:

$$K_{AB} = ((r_A + \pi(KT_{A1})d_A).l)(h.(KT_{B1} + \pi(KT_{B1})P_B))$$

và kiểm tra $MAC_K(2, KT_{A1}, KT_{B1})$

Tiếp đó A tính $MAC_K(3, KT_{A1}, KT_{B1})$và gửi cho B.

Bước 4. Kiểm tra (B2)

Thực thể B tính $MAC_K(3, KT_{A1}, KT_{B1})$ và kiểm tra thực thể A.

2.4. Hàm dẫn xuất khóa KDF

2.4.1. Hàm dẫn khóa 1

Dạng thức của KDF

$KDF(Z, OtherInput)$ trong đó $OtherInput$ là $keydatalen$ và $OtherInfo$ (các đại lượng này được giải thích về sau).

Các giá trị cố định:

$hashlen$: số nguyên chỉ độ dài đầu ra (theo bit) của hàm băm được sử dụng để dẫn xuất ra các khối của dữ liệu khóa bí mật.

$max_hash_inputlen$: số nguyên có giá trị là độ dài lớn nhất (theo bit) của (các) chuỗi bit đầu vào của hàm băm.

Các hàm bổ trợ:

H: là hàm băm được chấp thuận là hàm băm được quy định tại QCVN 5 : 2016/BQP Quy chuẩn kỹ thuật quốc gia về Chữ ký số sử dụng trong lĩnh vực ngân hàng.

Đầu vào:

Z: Chuỗi byte bí mật chia sẻ trước.

$keydatalen$: số nguyên chỉ độ dài (theo bit) của dữ liệu khóa bí mật được sinh ra; $keydatalen$ cần nhỏ hơn hoặc bằng $Hashlen \times (2^{32} - 1)$

$OtherInfo$: Chuỗi bit sau:

$$AlgorithmID||PartyAInfo||PartyBInfo\{||SuppPubInfo\}\{||SuppPrivInfo\}$$

trong đó các trường con được định nghĩa như sau:

- $AlgorithmID$: Chuỗi bit chỉ ra cách thức phân tách dữ liệu khóa đã được dẫn xuất ra và dữ liệu khóa được dẫn xuất sẽ được sử dụng cho những thuật toán nào. Ví dụ, $AlgorithID$ có thể chỉ ra rằng các bit 1-80 được dùng là 80-bit khóa cho $HMAC$ và các bit 81-208 được dùng là 128-bit khóa cho AES.

- $PartyAInfo$: Một chuỗi bit chứa các thông tin công khai được yêu cầu bởi ứng dụng sử dụng hàm KDF được đóng góp bởi bên A trong quá trình dẫn xuất khóa. Ở mức tối thiểu, $PartyAInfo$ chứa ID_A là định danh bên A. Xem chú ý phần dưới.

- *PartyBInfo:* Một chuỗi bit chứa các thông tin công khai được yêu cầu bởi ứng dụng sử dụng hàm KDF được đóng góp bởi bên B trong quá trình dẫn xuất khóa. Ở mức tối thiểu, $PartyVInfo$ chứa ID_B là định danh bên B. Xem chú ý phần dưới.

- (Tùy chọn) $SuppPubInfo$: Một chuỗi bit chứa các thông tin công khai bổ sung cả hai bên cùng biết (mutual-known)

- (Tùy chọn) $SuppPrivInfo$: Một chuỗi bit chứa thông tin bí mật bổ sung cả hai bên cùng biết (mutual-known)

(Ví dụ, một khóa bí mật đối xứng chia sẻ trước được truyền thông qua một kênh riêng biệt)

Thuật toán:

1. Tính $reps = \lceil keydatalen/hashlen \rceil$.

2. Nếu $(reps > 2^{32} - 1)$ thì ABORT: chỉ thị lỗi và dừng.

3. Khởi tạo bộ đếm chuỗi bit 32-bit big-endian *counter* bằng 00000001_{16}.

4. Nếu $counter||Z||OtherInfo$ có độ dài lớn hơn $max_hash_inputlen$ thì chỉ thị lỗi và dừng lại.

5. Vòng lặp với $i = 1$ đến $reps$, thực hiện:

5.1. Tính $Hash_i = H(counter||Z||OtherInfo)$.

5.2. Tăng $counter$ lên 1 ($modulo\ 2^{32}$), xử lý nó dưới dạng số nguyên không âm 32-bit.

6. Lấy $Hash$ là $Hash_{reps}$ nếu ($keydatalen/hashlen$) là số nguyên, ngược lại lấy số lượng ($keydatalen\ mod\ hashlen$) bit bên trái của đoạn dữ liệu $Hash_{reps}$.

7. Lấy $DerivedKeyingMaterial = Hash_1||\ Hash_2||....||Hash_{reps-1}||Hhash$

Đầu ra:

Chuỗi bit $DerivedKeyingMaterial$ có độ dài $keydatalen$ bit (hoặc thông báo lỗi). Thuật toán KDF tạo ra dữ liệu khóa có độ dài lớn nhất là $hashlen \times (2^{32} - 1)$. Bất kỳ lời gọi hàm KDF nào trong trường hợp sử dụng giá trị $keydatalen$ lớn hơn $hashlen \times (2^{32} - 1)$ sẽ dẫn tới chỉ thị lỗi và dừng mà không cho ra $DerivedKeyingMaterial$. Bất kỳ lời gọi hàm KDF nào dùng để băm một chuỗi bit có độ dài lớn hơn $max_hash_inputlen$ cũng sẽ dẫn tới thông báo lỗi và dừng mà không xuất ra $DerivedKeyingMaterial$.

Chú ý:

a) ID_A và ID_B sẽ được biểu diễn trong $OtherInfo$ là hai đơn vị thông tin riêng rẽ.

b) Bên A sẽ là bên khởi tạo và bên B có thể là bên trả lời của giao thức sử dụng lược đồ thỏa thuận khóa được dùng để xác định khóa bí mật chia sẻ trước Z.

2.4.2. Hàm dẫn xuất khóa 2

Ký hiệu $hashlen$ chỉ độ dài đầu ra của hàm hash được chọn và $maxhashlen$ *là* độ dài đầu vào cực đại của hàm hash.

Đầu vào

Đầu vào của hàm dẫn xuất khóa là

- Z Xâu bit là bí mật chia sẻ.

CHÚ THÍCH: Giao thức thỏa thuận khóa sử dụng mật mã trên đường cong elliptic dẫn xuất ra khóa bí mật chia sẻ K_{AB} hoặc dưới dạng một điểm trên đường cong elliptic hoặc dưới dạng ghép hai điểm. Trong trường hợp thứ nhất, để có được khóa mật chia sẻ Z làm đầu vào cho hàm dẫn xuất khóa thì phải áp dụng hàm π để chuyển điểm trên đường cong elliptic thành số nguyên và từ đó chuyển sang xâu bit. Trong trường hợp thứ hai áp dụng hàm π cho cả hai điểm để được hai số nguyên z_1, z_2; hai số nguyên này sau đó được biến đổi thành các xâu bít và được ghép lại với nhau.

- *Keydatalen* Số nguyên biểu thị độ dài tính bằng bit của dữ liệu khóa được tạo ra, nhỏ hơn đại lượng $hashlen \times (2^{32} - 1)$

- (Tùy chọn) Xâu bit $SharedInfor$ gồm một dữ liệu nào đó được hai thực thể dùng chung nhằm chia sẻ bí mật Z.

Thuật toán

Hàm dẫn xuất khóa được tính như sau:

1. Khởi động bộ đếm 32-bit 00000001 (Hệ thập lục)

2. Vòng lặp với $i = 1$ đến $j = \lceil keydatalen/hashlen \rceil$ thực hiện:

- Tính $Hash_i = H(Z \parallel counter \parallel SharedInfo)$

3. Tăng giá trị bộ đếm

4. Tăng i

5. Giả sử $HHash_j$ biểu thị $HHash_j$ khi $Keydatalen/hashlen$ là số nguyên và biểu thị $(Keydatalen(hashlen \times (j-1)))$ bit bên trái nhất của $Hash_i$ trong trường hợp ngược lại

6. Đặt $Keydata = Hash_1 \parallel Hash_2 \parallel \cdots \parallel Hash_{j-1} \parallel HHash_j$

Đầu ra

Là dữ liệu khóa ở dạng xâu bit có độ dài bằng $keydatalen$.

CHÚ THÍCH: Lưu ý là hàm dẫn xuất khóa tạo ra dữ liệu khóa có độ dài nhỏ hơn $hashlen \times (2^{32}-1)$ bit. Bất kỳ lược đồ nào gọi hàm dẫn xuất khóa cho xâu bít lớn hơn hoặc bằng $hashlen \times (2^{32}-1)$ bit sẽ cho ra thông báo "lỗi" và dừng lại. Tương tự, tất cả hàm dẫn xuất khoá được gọi ra không băm các xâu bít có độ dài lớn hơn $maxhashlen$. Bất kỳ lược đồ nào gọi hàm dẫn xuất khóa băm các xâu bít có độ dài lớn hơn $maxhashlen$ đều cho ra thông báo "lỗi" và dừng lại.

2.5. Giao thức vận chuyển khóa bí mật

Điều này trình bày giao thức vận chuyển khóa bí mật, một khóa được truyền từ thực thể A sang thực thể B và một khóa được truyền từ thực thể B sang thực thể A.

Bước 1. Kiến thiết thẻ khóa (A1): Thực thể A tạo ra một thẻ khóa KT_{A1} bao gồm một số ngẫu nhiên r_A và một trường dữ liệu tùy chọn $Text1$ rồi gửi nó cho thực thể B:

$$KT_{A1} = r_A \parallel Text1$$

Bước 2. Mã khóa khối khóa (B1.1): Thực thể B có một khóa K_B và muốn gửi một cách an toàn cho thực thể A. Trước hết B tạo ra khối dữ liệu khóa bao gồm định danh riêng biệt của bên gửi B, khóa K_B và trường dữ liệu tùy chọn $Text2$. Thực thể B mã hóa khối dữ liệu khóa này bằng phép mã công khai E_A của A, thu được khối mã:

$$BE_1 = E_A(ID_B \parallel K_B \parallel Text2)$$

Bước 3. Kiến thiết thẻ khóa (B1.2): Thực thể B tạo ra một khối thẻ khóa bao gồm định danh riêng biệt của bên nhận A, một số ngẫu nhiên r_A nhận được ở bước 1, một số ngẫu nhiên mới r_B (tùy chọn) do B tạo ra, khối đã mã BE_1 và một trường dữ liệu tùy chọn $Text3$. Tiếp đó, B tiến hành ký khối dữ liệu thẻ bằng phép ký bí mật của mình và gửi kết quả cho A:

$$KT_{B1} = S_B(r_B \parallel r_A \parallel ID_A \parallel BE_1 \parallel Text3) \parallel Text4$$

Bước 4. Kiểm tra thẻ khóa (A2.1): Thực thể A sử dụng phép kiểm tra công khai của bên gửi V_B để kiểm tra chữ ký số của thẻ khóa nhận được KT_{B1}. Tiếp đó A kiểm tra định danh riêng biệt A và kiểm tra giá trị nhận được r_A xem có khớp với số ngẫu nhiên nhận được ở Bước 1 (A1) hay không.

Bước 5. Giải mã khối khóa (A2.2): Thực thể A tiến hành giải mã khối BE_1 bằng phép giải mã bí mật D_A của mình. Tiếp đó, A kiểm tra định danh riêng biệt của bên gửi B. Nếu tất cả lần kiểm tra đều thành công thì A chấp nhận khóa K_B.

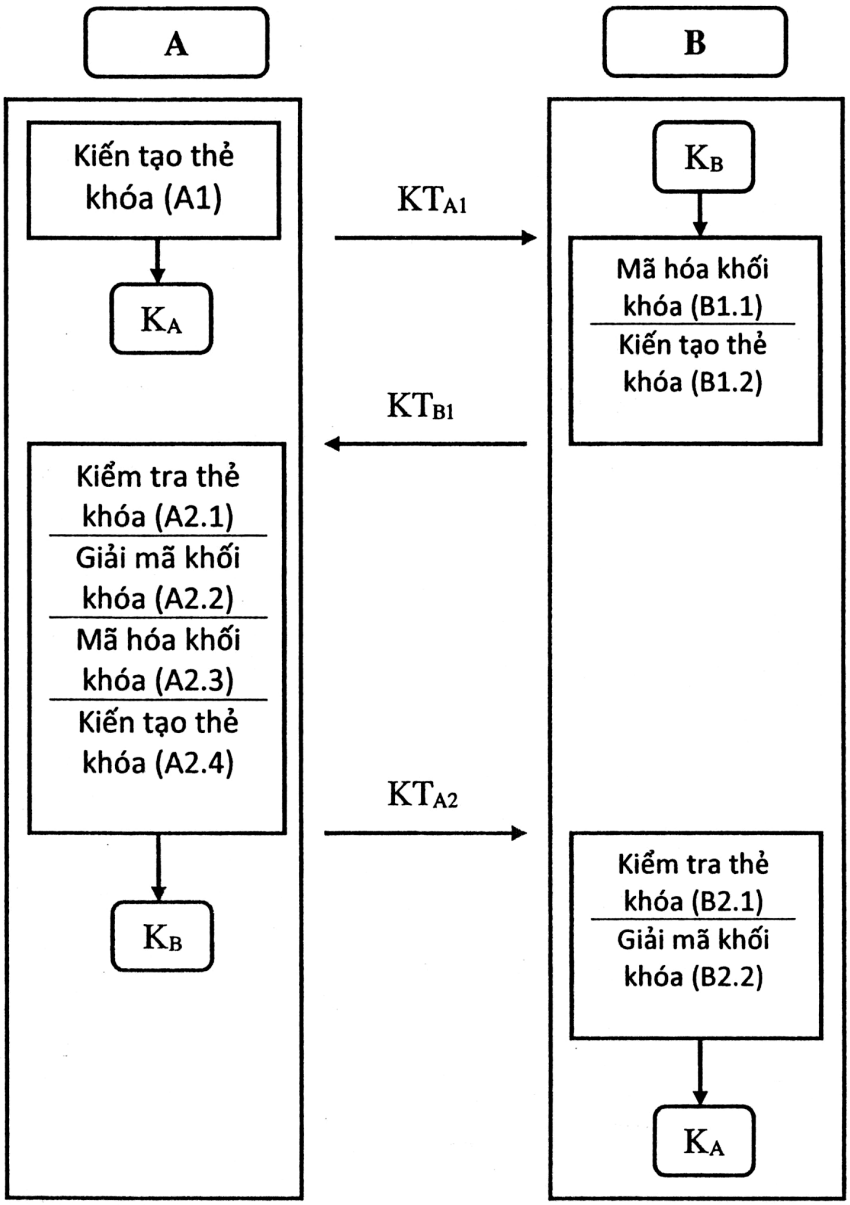
Bước 6. Mã khóa khối khóa (A2.3): Thực thể A có một khóa K_A muốn gửi cho B một cách an toàn. Trước hết, A tạo ra khối dữ liệu khóa bao gồm định danh riêng biệt của bên gửi A, khóa K_A và trường dữ liệu tùy chọn $Text5$. Tiếp đến, A mã hóa khối dữ liệu khóa này bằng phép mã công khai E_B của B để được khối mã:

$$BE_2 = E_B(ID_A \parallel K_A \parallel Text5)$$

Bước 7. Kiến thiết thẻ khóa (A2.4): Thực thể A tạo ra một khối thẻ khóa bao gồm định danh riêng biệt của bên nhận B, một số ngẫu nhiên r_A do A tạo ra bước 1 (A1), một số ngẫu nhiên mới r_B do B tạo ra ở (B1.2), khối đã mã BE_2 và một trường dữ liệu tùy chọn $Text6$. Tiếp đó, A tiến hành ký khối dữ liệu thẻ bằng phép ký bí mật của mình và gửi kết quả cho B:

$$KT_{A2} = S_A(r_A \parallel r_B \parallel ID_B \parallel BE_2 \parallel Text6) \parallel Text7$$

Bước 8. Kiểm tra thẻ khóa (B2.1) Thực thể B sử dụng phép kiểm tra công khai của bên gửi V_A để kiểm tra chữ ký số của thẻ khóa nhận được KT_{A2}. Tiếp đó B kiểm tra định danh riêng biệt B của mình và kiểm tra giá trị nhận được r_B xem có khớp với số ngẫu nhiên ở Bước 3 (B1.2) hay không. Ngoài ra, B cũng kiểm tra giá trị ngẫu nhiên nhận được r_A xem có khớp với số ngẫu nhiên ở Bước 1 (A1) hay không.



Hình 2 – Cơ chế vận chuyển khóa bí mật

Bước 9. Giải mã khối khóa (B2.2): Thực thể B tiến hành giải mã khối BE_2 bằng phép giải mã bí mật D_B của mình. Tiếp đó, B kiểm tra định danh riêng biệt của bên gửi A. Nếu tất cả phép kiểm tra đều thành công thì B chấp nhận khóa K_A.

CHÚ THÍCH: Trong tình huống chỉ có một bên gửi khóa bí mật cho bên kia, chẳng hạn thực thể B muốn chuyển khóa bí mật K_B cho A, thì chỉ cần thực hiện giao thức trên từ bước 1 đến bước 5.

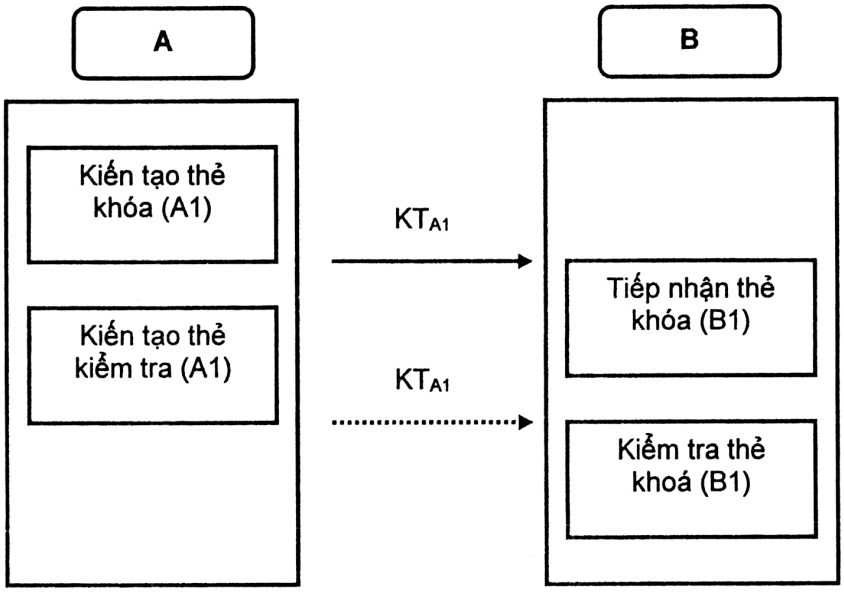
2.6. Giao thức vận chuyển khóa công khai

Điều này trình bày hai giao thức truyền thông tin khóa công khai từ một thực thể A đến một thực thể B. Giao thức thứ nhất (giao thứ 2.6.1) không sử dụng bên thứ ba tin cậy, theo giao thức này để kiểm tra tính toàn vẹn và nguồn gốc của thông tin khóa công khai, hai thực thể sử dụng một hàm băm được quy định tại QCVN 5 : 2016/BQP. Giao thức thứ hai (giao thức

2.6.2) giả thiết là chứng thư khóa công khai hợp lệ $Cert_A$ của A được cấp bởi bên thứ ba tin cậy là Tổ chức chứng thực và cấp chứng chỉ khóa công khai CA, theo giao thức này thực thể B có thể truy cập vào bản sao có xác thực của phép kiểm tra khoá công khai của tổ chức này.

2.6.1 Giao thức vận chuyển khóa công khai không sử dụng bên thứ ba tin cậy

Giao thức gồm bốn bước và được minh họa trên Hình 3.



Hình 3: Cơ chế vận chuyển khóa công khai

Bước 1. Kiến thiết thẻ khóa (A1): A tạo ra một thẻ khóa KT_{A1} bao gồm thông tin khóa công khai của A và gửi đến cho B:

$$KT_{A1} = PKI_A \parallel Text1$$

Bước 2. Tiếp nhận thẻ khóa (B1): B tiếp nhận được thẻ khóa, trích lấy thông tin khóa công khai PKI_A. Hoặc B sẽ thực hiện kiểm tra khóa kiểm tra của A hoặc sẽ lưu trữ nó ở nơi tránh được giả mạo phục vụ để cho lần kiểm tra sau hoặc sẽ sử dụng.

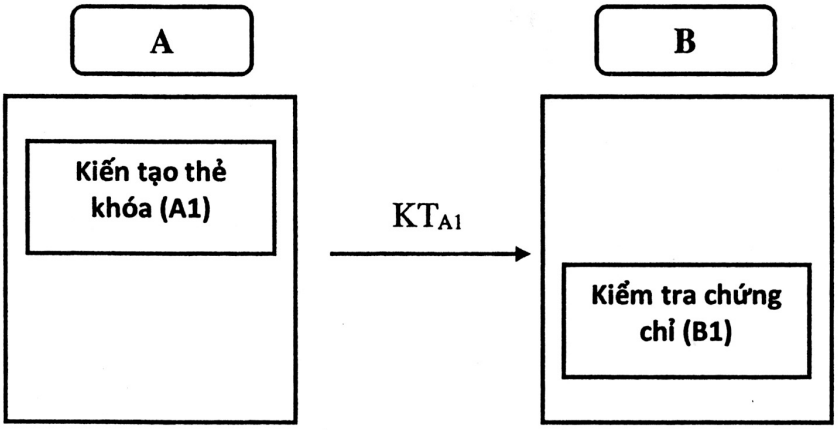
Bước 3. Kiến tạo thẻ kiểm tra (A2): A tính toán giá trị kiểm tra $hash(PKI_A)$ đối với thông tin khóa công khai của mình và gửi giá trị kiểm tra này cùng với các định danh tùy chọn riêng biệt của A và B rồi gửi đến thực thể B sử dụng một kênh truyền có xác thực và độc lập thứ hai (ví dụ kênh truyền điện báo hoặc đường truyền thư đăng ký trước).

$$KT_{A2} = A \parallel B \parallel hash(PKI_A) \parallel Text2$$

Bước 4. Kiểm tra thẻ khóa (B2): Dựa vào thông tin thẻ khóa nhận được KT_{A2}, B có thể tùy chọn kiểm tra định danh riêng biệt của A và B, tính toán ra giá trị kiểm tra trên thông tin khóa công khai của A nhận được từ thẻ khóa KT_{A1} và so sánh với giá trị kiểm tra nhận được từ thẻ khóa KT_{A2}. Nếu kết quả kiểm tra thành công thì B lấy khóa công khai của A đưa lên danh sách các khóa đang hoạt động (danh sách này được bảo vệ chống lại sự giả mạo).

2.6.2 Giao thức vận chuyển khóa công khai sử dụng bên thứ ba tin cậy

Giao thức gồm hai bước và được minh họa trên Hình 4.



Hình 4 – Cơ chế vận chuyển khóa công khai 2

Bước 1. Kiến thiết thẻ khóa (A1): Thực thể A tạo ra một thẻ khóa KT_A bao gồm chứng chỉ khóa công khai của A và gửi nó cho B:

$$KT_A = Cert_A \parallel Text$$

Bước 2. Kiểm tra chứng chỉ (B1): Dựa trên thông tin nhận được về chứng chỉ khóa công khai, B sử dụng phép kiểm tra công khai V_{CA} của CA để kiểm tra tính xác thực của thông tin khóa công khai và kiểm tra cả tính hợp lệ đối với khóa công khai của A.

2.7. Quy định kỹ thuật cho các tham số

2.7.1. Quy định về nguồn ngẫu nhiên

Các số ngẫu nhiên được sử dụng cho các mục đích khác nhau như để sinh các tham số mật mã, các khóa mật mã, các giá trị ngẫu nhiên dùng một lần và các giá trị thách đố xác thực.

Một số bộ sinh bit ngẫu nhiên tất định DRBG được chấp thuận để sử dụng theo quy định chung bao gồm: HASH_DRBG, HMAC_DRBG và CTR_DRBG.

Các bộ sinh bit ngẫu nhiên RBG tuân theo SP800-90A phiên bản sửa đổi lại năm 2015 để sinh bit ngẫu nhiên cũng được chấp thuận để sử dụng tiếp.

2.7.2. Quy định đối với tham số RSA

2.7.2.1. Các yêu cầu chung

1. Số mũ công khai e cần phải được chọn với các ràng buộc sau:

a) Số mũ công khai e cần được chọn trước khi tạo số mũ bí mật d;

b) Số mũ công khai e cần phải là số nguyên dương lẻ sao cho

$$65{,}537 \leq e < 2^{nlen-2security_strength}$$

Với $nlen$ là độ dài của modulo n theo bit.

Chú ý rằng e có thể là giá trị bất kỳ thoả mãn ràng buộc 1(b); p và q sẽ được chọn (trong mục 2) sao cho e là nguyên tố cùng nhau với cả $(p-1)$ và $(q-1)$.

2. Hai số nguyên tố p và q được tạo ngẫu nhiên và giữ bí mật cần phải được chọn với các ràng buộc sau:

a) $(p-1)$ và $(q-1)$ cần phải nguyên tố cùng nhau với số mũ công khai *e*;

b) Mỗi một trong bốn số $(p+1)$, $(p-1)$ và $(q+1)$, $(q-1)$ cần phải có các nhân tử nguyên tố lớn hơn $2^{security_strength+20}$;

c) Nhân tử nguyên tố bí mật p, q cần phải được chọn ngẫu nhiên từ các số nguyên tố thỏa mãn $(\sqrt{2})(2^{(nlen/2)-1}) \leq q < p \leq 2^{(nlen/2)} - 1$;

d) $|p-q| > 2^{(nlen/2-100)}$.

3. Số mũ bí mật d cần phải được lựa chọn sau khi tạo p và q với các ràng buộc:

a) Số mũ *d* cần phải lớn hơn $2^{(nlen/2)}$, và

b) $d = e^{-1} mod\ (LCM((p-1),(q-1)))$

(Chi tiết về hàm tạo các tham số RSA có thể tham khảo trong tài liệu *FIPS 186-4: Digital Signature Standard*)

2.7.2.2. Quy định ngưỡng giá trị cho tham số theo thời hạn sử dụng

Thời hạn Quy định	RSA
2020	$\|p\| = 1536$
2030	$\|p\| = 2048$
Sau 2030	$\|p\| = 3072$

2.7.3. Hệ mật dựa trên Logarit rời rạc DL

Tham số của hệ mật dựa trên bài toán Logarit rời rạc trên trường hữu hạn F_p là bộ:

(p,q,g), trong đó p là đặc số của trường F_p, q là bậc của nhóm $F^*(p)$ và là ước của $p-1$ thoả mãn các điều kiện sau:

- p,q là số nguyên tố

- Độ dài của p,q được cho dưới bảng sau:

Năm	Độ dài p	Độ dài q
Đến 2020	$\|p\| = 1536$	$\|q\| = 192$
Đến 2030	$\|p\| = 2048$	$\|q\| = 224$
Sau năm 2030	$\|p\| = 3072$	$\|q\| = 256$

2.7.4. Hệ mật ECC

2.7.4.1. Quy định về các khóa

Kiểm tra tính hợp lệ của các tham số miền $(p, SEED, a, b, G, n, h)$ như sau:

Xâu $SEED$ dùng để sinh ngẫu nhiên đường cong Elliptic xác định trên trường F_p với p là số nguyên tố lẻ.

Trước khi sử dụng một bộ tham số miền, tính hợp lệ của nó phải được kiểm tra theo thuật toán dưới đây:

1. Kiểm tra p là một số nguyên tố lẻ.
2. Kiểm tra a, b, x_G, y_G là các phần tử của trường F_p.
3. Kiểm tra rằng a và b được dẫn xuất tương ứng từ $SEED$.
4. Kiểm tra $(4a^3 + 27b^2)$ khác 0 và $j(E) \neq 0; 1728$ trong F_p
5. Kiểm tra $y_G^2 = x_G^3 + ax_G + b$ trong F_p.
6. Kiểm tra n là nguyên tố và $n > 4\sqrt{p}$.
7. Kiểm tra $nG = O_E$.
8. Kiểm tra đường cong có thuộc danh sách các đường cong yếu:
   a. Thỏa mãn điều kiện MOV, (chú ý rằng một đường cong thỏa mãn điều kiện MOV sẽ không phải là đường cong siêu biến)
   b. Kiểm tra đường cong không kỳ dị, nghĩa là $\#E \neq p$.

Nếu bất kỳ sự kiểm tra nào ở trên thất bại thì tham số miền phải được xem là không hợp lệ.

Điều kiện MOV được hiểu là không có giá trị k nguyên dương nào $0 < k < B$ để cho $p^k - 1$ chia hết cho n. Trên thực hành hiện nay $|p| = 224$ bit thì người ta xét với $B = 15$ là đủ vì khi đó $|p^k| = 3360 > 2048$.

Các hệ số a, b của đường cong được sinh ngẫu nhiên trên F_p từ đầu vào *SEED* và có thể kiểm tra được.

Khóa bí mật d phải được sinh ngẫu nhiên trong khoảng $[1, n - 1]$.

Đường cong Elliptic xác định trên trường hữu hạn F_p với tối thiểu $|p| = 224$ bit và được xác định cụ thể như sau:

Độ dài bit của n	**Độ dài bit của p**
224 - 255	|p| = 224
256 - 383	|p| = 256
384 - 511	|p| = 384
≥ 512	|p| = 521

Đại lượng Cofactor được định nghĩa và ký hiệu là $h = \#E(F_p)/n$ tuân theo bảng dưới đây:

Độ dài bit của n	**Giá trị h cực đại cho phép**
224 - 255	2^{14}

256 - 383	2^{16}
384 - 511	2^{24}
≥ 512	2^{32}

2.7.5. Độ an toàn theo bit quy đổi giữa RSA, DL và ECC

Độ an toàn	ECC	RSA	DL
112	224	2048	2048
128	256	3072	3072
192	384	7680	7680
256	512	15360	15360

3. QUY ĐỊNH VỀ QUẢN LÝ

3.1. Các mức giới hạn của đặc tính kỹ thuật mật mã và yêu cầu quản lý khóa mật mã nêu tại Quy chuẩn này là các chỉ tiêu chất lượng phục vụ được quản lý theo quy định về quản lý chất lượng sản phẩm, dịch vụ mật mã dân sự được quy định tại Luật An toàn thông tin mạng ngày 19 tháng 11 năm 2015.

3.2. Hoạt động kiểm tra chất lượng sản phẩm mật mã được cơ quan quản lý nhà nước có thẩm quyền tiến hành định kỳ hàng năm hoặc đột xuất.

3.3. Quản lý sử dụng khóa mật mã:

- Các loại khóa mật mã phải được lập thành danh mục với việc mô tả chi tiết mục đích sử dụng, thời hạn sử dụng.
- Đảm bảo sử dụng khóa đúng mục đích được quy định (chẳng hạn khóa dùng để mã các khóa khác không được sử dụng để mã hóa dữ liệu), không sử dụng khóa đã hết hạn sử dụng hoặc hủy khóa trước thời hạn đảm bảo khả năng giải mã khi dữ liệu hết thời hạn bảo mật. Không cung cấp khóa cho người nhận không có thẩm quyền.
- Khóa mật mã phải được bảo vệ chống lại các mối nguy cơ như bị lộ (ngoại trừ các khóa công khai), sửa đổi, phá hủy và tái sử dụng. Việc bảo vệ phải thực hiện suốt cả vòng đời của khóa. Mục tiêu bảo vệ phải phù hợp với từng loại khóa. Đối với khoá công khai cần đảm bảo tính toàn vẹn và tính khả dụng, đối với khóa bí mật phải đảm bảo đầy đủ cả ba tính chất: bí mật, toàn vẹn và khả dụng. Bảo vệ khóa phải kết hợp đồng bộ các giải pháp bằng kỹ thuật mật mã, phương tiện vật lý và phương tiện tổ chức để tạo ra một vùng an toàn để cất giữ khóa, sử dụng khóa và thực thi thuật toán mật mã:
  + Khóa phải được lưu trữ trong một thiết bị lưu trữ an toàn tách biệt (được gọi là vật mang khóa như HSM, thẻ Token hay thẻ thông minh) để đảm bảo khoá được lưu trữ và tương tác với các thuật toán mật mã an toàn. Trong trường hợp vật mang khóa là mô-đun an toàn phần cứng HSM, yêu cầu an toàn tối thiểu mức 3, trường hợp vật mang khóa là thẻ Token hay thẻ thông minh, yêu cầu an toàn tối thiểu mức 2 (mức an toàn được quy định tại TCVN 11295:2016 (ISO/IEC 19790:2012) Công nghệ thông tin – Các kỹ thuật an toàn - Yêu cầu an toàn cho mô-đun mật mã).
  + Trường hợp khóa lưu trữ ở dạng mềm cần sử dụng kỹ thuật mật mã để bảo vệ (Tuân thủ theo các quy định tại QCVN 4:2016/BQP và QCVN 5:2016/BQP).
  + Trường hợp hệ thống khóa được tổ chức theo cấu trúc phân cấp, các khóa cùng tầng chỉ được dùng để bảo vệ khóa ở tầng kế tiếp. Khóa chủ, tức khóa ở tầng cao nhất, phải được bảo vệ bằng cách chia ra nhiều thành phần và mỗi thành phần được bảo vệ riêng. Biện pháp này nhằm chống nguy cơ khóa bị lộ từ bên trong nội bộ.
- Việc hủy khóa phải đảm bảo hủy bỏ được tất cả các bản ghi khóa, bao gồm cả các bản sao dự phòng, sao cho không thể khôi phục được bất kỳ thông tin khóa nào dù bằng bất kỳ phương tiện nào. Đồng thời không để bất kỳ khóa nào bị hủy bỏ trước thời hạn nhằm đảm bảo khả năng giải mã khi dữ liệu hết thời hạn bảo mật.
- Nếu khóa bị tổn thương hoặc bị nghi ngờ tổn thương hoặc có sự thay đổi chủ sở hữu thì phải thực hiện thu hồi khóa để vô hiệu hóa nguy cơ bị lộ hoặc mất an toàn.
- Thiết bị lưu trữ khóa phải được quản lý bởi người sử dụng. Người sử dụng có trách nhiệm bảo quản khóa ở vị trí an toàn, chống tiếp cận trái phép, chống sao chép hoặc bị đánh cắp. Không cung cấp khóa cho người không có thẩm quyền.

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

4.1. Các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự có trách nhiệm đảm bảo tuân thủ Quy chuẩn này và chịu sự kiểm tra của cơ quan quản lý nhà nước theo quy định.

4.2. Doanh nghiệp cung cấp sản phẩm, dịch vụ mật mã dân sự cho các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) có trách nhiệm thực hiện công bố hợp quy sản phẩm, dịch vụ mật mã dân sự phù hợp với Quy chuẩn này. Việc công bố hợp quy thực hiện theo Thông tư số 28/2012/TT-BKHCN ngày 12 tháng 12 năm 2012 của Bộ Khoa học và Công nghệ.

4.3. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ có trách nhiệm tiếp nhận đăng ký công bố hợp quy, thực hiện quản lý, hướng dẫn và kiểm tra việc công bố hợp quy.

5. TỔ CHỨC THỰC HIỆN

5.1. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ có trách nhiệm hướng dẫn, tổ chức triển khai quản lý kỹ thuật mật mã của Quản lý khóa theo Quy chuẩn này.

5.2. Trong trường hợp các quy định nêu tại Quy chuẩn kỹ thuật quốc gia này có sự thay đổi, bổ sung hoặc được thay thế thì thực hiện theo quy định tại văn bản mới./.